Lab SL04
WebSphere Application Server for iSeries V5: Configuration and Administration

ITSO iSeries Technical Forum 2003

Presented by
Aleksandr Nartovich

Contents

Lab 1

WebSphere installation

In this exercise, you configure a separate WebSphere instance on the iSeries server, which you will use throughout the course.

Objectives

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

- Use OS/400 Qshell to run WebSphere script commands
- Create a new WebSphere Application Server instance
- Start a WebSphere Application Server instance
- Verify that an instance is started and operational

Lab environment

Be sure that you lab environment meets the following prerequisites:

- Processor: Pentium III 300 MHz or better
- 384 MB memory or better
- 4 GB disk space

Note: These requirements exceed the minimum requirements for the product because you will perform more complex configuration than for a basic installation.

The following software should already be installed:

- Microsoft Windows 2000
- Microsoft Internet Explorer 5.0 or higher
- Adobe Acrobat Reader 3.0 or later

TCP/IP networking should be correctly configured. Make sure you know the host name of the iSeries server, because you need it in the exercises.

In this exercise, the WebSphere Application Server (Base) software and associated PTFs are already installed. You create a new application server.

Time required

The time required to efficiently complete this lab project is 30 minutes.

Creating an application server

In this lab, you create a new instance of the WebSphere Application Server Version 5. You do this, so that each team can operate in their own unique WebSphere environment. This allows each team to configure and use resources independent of one another. This is useful in other scenarios, such as having multiple versions of the same application running simultaneously on the same iSeries server for development purposes.

In WebSphere Application Server Version 5, you can create, start, and stop the application server independently of creating the HTTP administrative server instance.

As you complete this lab, you need to refer to the workshop cover sheet that contains the system name, user ID, password, and so on.

Setting up a new WebSphere Application Server instance requires the following steps:

- Running script commands in the Qshell Interpreter
- Creating a new WebSphere Application Server instance
- Starting a new instance of WebSphere
- Verifying that the server environment is started and operational

To save time and allow you to explore more functions in this exercise, we already installed the WebSphere Application Server (Base) product. The default application server name is named Server1. The new server name that you create will be named TeamxSvr. Throughout these lab exercises, replace x with your team number.

Important: Instance and server names are case sensitive.

The actual exercise instructions starts with Task 1:, “Creating a new WebSphere Application Server instance” on page 3.

Creating a new WebSphere Application Server instance

WebSphere Application Server provides several Qshell scripts to help you to create and manage WebSphere Application Server instances. The WebSphere Application Server Version 5 product includes a default instance that consists of a single application server. The application server, named Server1, contains the administrative console application, a default application, and the install verification application.

Using multiple instances allows you to create multiple WebSphere Application Server environments that are completely isolated from one another. This is the approach that we use in this class.

You can run your applications in any of these environments. In WebSphere Application Server (Base) new servers are created using the Qshell Interpreter on the iSeries server.

The instance management scripts are shipped in the WAS_INSTALL_ROOT/bin directory where WAS_INSTALL_ROOT is `/QIBM/ProdData/WebAS5/Base` for WebSphere Application Server (Base).

Task 1: Creating a new WebSphere Application Server instance

Important: THIS IS READ-ONLY EXERCISE.

To create a new instance, run the crtwasinst script from the Qshell command line on the iSeries server hosting your server. This script creates all new server directories and sets up the correct authorities. To run this script, your iSeries user profile must have *ALLOBJ authority:

__ 1. Sign on to a session on the iSeries server using your team ID and password

__ 2. On the OS/400 command line, enter the STRQSH (Start Qshell) command.

__ 3. On the Qshell command line, use the `cd` command to change to the directory that contains the create instance script, for example:

```
cd /QIBM/ProdData/WebAS5/Base/bin
```

__ 4. Run the **`crtwasinst`** script:

```
crtwasinst -instance instance -server serverName -portblock 6x01 -exthttp 6x00
```

For example for Team1:

```
crtwasinst -instance TeamX -server TeamXSvr -portblock 6101 -exthttp 6100
```

Here *instance* is the name of the instance that is created (for this exercise the name is Teamx, where *x* is your team number). *serverName* provides the server's name and *portblock* is the first number of a block of ports that your instance will use.

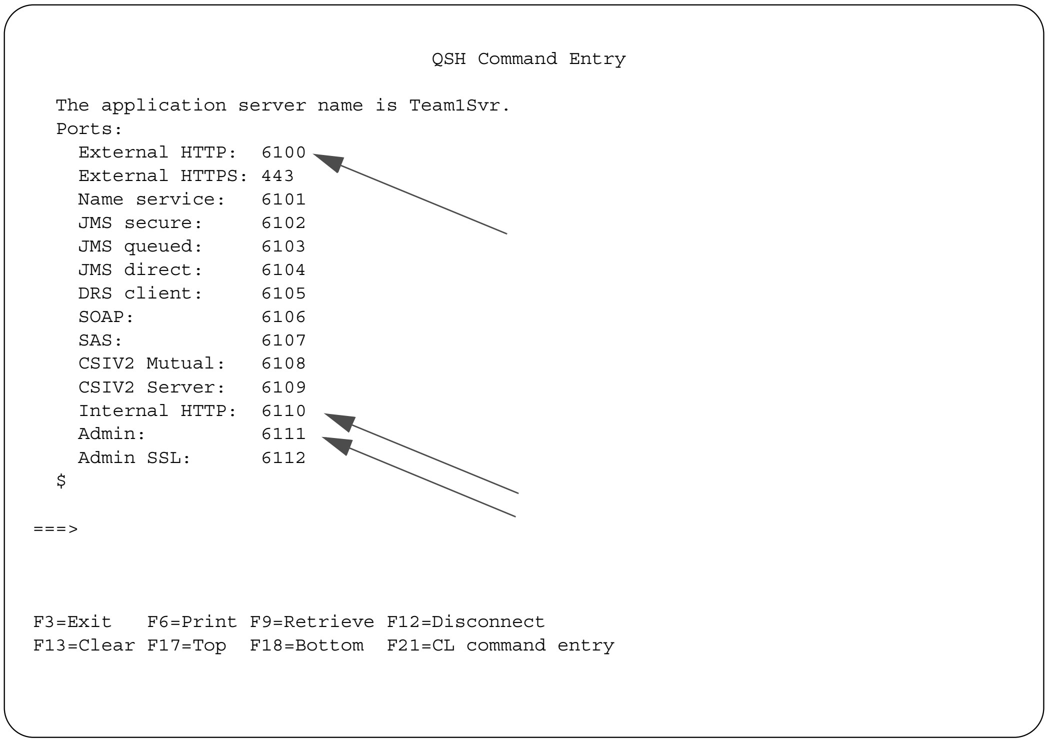
The value *exthttpport* specifies the number of the TCP/IP port where the external HTTP server listens. The default value is 80. You must configure the external HTTP server through its administrative interface, so that WebSphere Application Server also listens on this port. The default is port 80. However, this is the port address used by the default server, Server1. Therefore, we must assign our individual external HTTP server instances a different unused address.

Note: During this workshop, we work in teams. The teams are numbered 0 through 9 depending on the size of the class. Your *portblock* will start with 6x01, where x is your team number. In addition, we use the *exthttp* parameter to specify the external HTTP port address. This number for your team should be 6x00, where x is again your team number.

You can use the Work with TCP/IP Network Status (NETSTAT *CNN) command to display a list of port numbers that are currently being used.

Notes: Although *exthttpport* and *portblock* are optional parameters, we recommend that you always specify these parameters if you do not want your instance's ports to conflict with the default instance's ports.

__ 5. Figure 1-1 shows an example of the results of running the **`crtwasinst`** script using the portblock parameter.



```
                                      QSH Command Entry

  The application server name is Team1Svr.
  Ports:
    External HTTP:  6100
    External HTTPS: 443
    Name service:   6101
    JMS secure:     6102
    JMS queued:     6103
    JMS direct:     6104
    DRS client:     6105
    SOAP:           6106
    SAS:            6107
    CSIV2 Mutual:   6108
    CSIV2 Server:   6109
    Internal HTTP:  6110
    Admin:          6111
    Admin SSL:      6112
  $

===>



F3=Exit   F6=Print F9=Retrieve F12=Disconnect
F13=Clear F17=Top  F18=Bottom  F21=CL command entry
```

Figure 1-1 crtwasinst sample screen output

Take note of the output display from the results of the crtwasinst command.

What is the port assignment for the administrative console? Write down this port number (replace X with your team number):

Admin Console port number (displayed as *Admin*) ________

(where your port number look like 6x11)

Keep track of the external and internal HTTP port addresses. You need this information in a later lab:

Internal HTTP port _______.

(where your port number look like 6x10)

External HTTP port _______.

(where your port number look like 6x00)

Task 2: Exploring the WebSphere Application Server directory structure

Follow these steps:

__ 1. Start Windows Explorer and locate the IFS of your iSeries server. It is either mapped as a network drive or in the Network Neighborhood.

Note: if none of the drives mapped to IFS, you will have to do this in order to continue the lab. These are the methods to map a network drive:

- Select **Search -> For Files or Folders** from the Windows Start menu. Click **Computers** link in the left navigation pane and, in the computer name field, enter the name of your iSeries server. When the server is found, double-click on it and supply a valid user name and password to access it. After this, you should be exploring the server's integrated file system which contains the root share. Right-click the share and select **Map Network Drive**. Choose an available drive letter for the share, for example X:

- In Windows Explorer select **Tools -> Map Network Drive** and choose **Map Network Drive**. In the resulting dialog box, choose a drive letter to map to and enter **\\<iSeries server name>\root** in the path field. Then supply a valid user name and password.

__ 2. Expand the **/QIBM/ProdData/WebAS5/Base** directory. All WebSphere Application Server instances are created under this directory structure. This directory holds production data, such as the executables, product properties, and so on

__ 3. Look at the subdirectories. The important subdirectories are:

bin	Programs, scripts, xml files, DTDs, etc.
httpAdmin	HTTP Admin JAR file
lib	Product JAR files
properties	messages, schemas, version information
samples	Test applications for development
web	Online help and documentation

__ 4. Expand the **/QIBM/UserData/WebAS5/Base** directory. This directory holds user data, such as Web applications and application server instances.

__ 5. Look at the subdirectories. There is one for each application server instance. Locate the subdirectory that represents your team instance and look at its subdirectories. The important subdirectories are:

config	xml.cnfg files, plugin-cfg.xml, node information
logs	Log and trace files
properties	Configuration properties for the server
wstemp	Temporary file for files created during JSP processing

Task 3: Starting the new application server instance

To start an instance, run the startServer script from the Qshell command line:

__ 1. On the OS/400 command line, run the STRQSH (Start Qshell) command.

__ 2. On the Qshell command line, use the `cd` command to change to the directory that contains the script, for example:

```
cd /QIBM/ProdData/WebAS5/Base/bin
```

__ 3. Run the startServer script:

```
startServer -instance instance serverName
```

where *instance* is `teamx` and *serverName* is `TeamxSvr` (substitute x with your team number)

For example:

```
startServer -instance Team1 Team1Svr
```

Here *instance* is the name of the instance you want to start (Team1).

Task 4: Verifying that the application server has started

Before you start the administrative console (which we do in the next lab exercise), verify that the environment started successfully. When the WebSphere Application Server environment is ready for use, a message is written to the job log of the application server job indicating that the WebSphere Application Server environment is ready: this is the most common way. Alternatively, you can wait till the script has finished. It displays the access port for the administrative console.

To determine if the WebSphere Application Server environment is ready, perform these steps from an OS/400 command line:

__ 1. Run the Work with Active Jobs (WRKACTJOB) command, specifying the appropriate subsystem on the subsystem (SBS) parameter. For WebSphere Application Server (Base), use the QEJBAS5 subsystem:

```
WRKACTJOB SBS(QEJBAS5)
```

__ 2. Find your application server job (*Team<u>x</u>Svr*, where *x* is your team number).

__ 3. Specify option `5` (Work with Job) on the option line next to the job.

__ 4. Press Enter.

__ 5. On the command line, specify option `10` (Display job log).

__ 6. Press Enter.

__ 7. Press F10

__ 8. Look for this message:

```
WebSphere application server application_server ready.
```

Here *application_server* is the name of your application server.

If the message is not displayed, press F5 to refresh the job log messages until the message is displayed. When the message is displayed, the WebSphere Application Server environment has successfully started. It may take up to 20 minutes for the message to be displayed, depending on your iSeries server. If the message is not displayed, see your instructor.

__ 9. To display the port number on which the application server is listening for the administrative console, position the cursor on the last line of the message.

__ 10. Press F1. This message is displayed:

```
WebSphere application server application_server in job app_server_job is ready to
handle administrative requests on port port_number
```

Here *application_server* is the name of your application server, *app_server_job* is the OS/400 job name for your application server, and *port_number* is the number of the port used by the administrative console.

__ 11. Press F3 twice to exit.

Task 5: Connecting to the admin console

__ 1. After the application server job has been started, you can connect to the WAS admin console. The admin console in WAS V5 is implemented as a Web application running in WAS - this is different from the previous versions.

__ 2. In Microsoft Internet Explorer 5.0, and 5.5, perform the following steps:

i. Click the **Tools** pull-down menu, and choose **Internet Options**.
ii. Click the **Security** tab.
iii. Click **Custom Level**. Under Cookies, select **Enable**.
iv. Click **OK**.

__ 3. In Microsoft Internet Explorer 6.0, perform these steps:

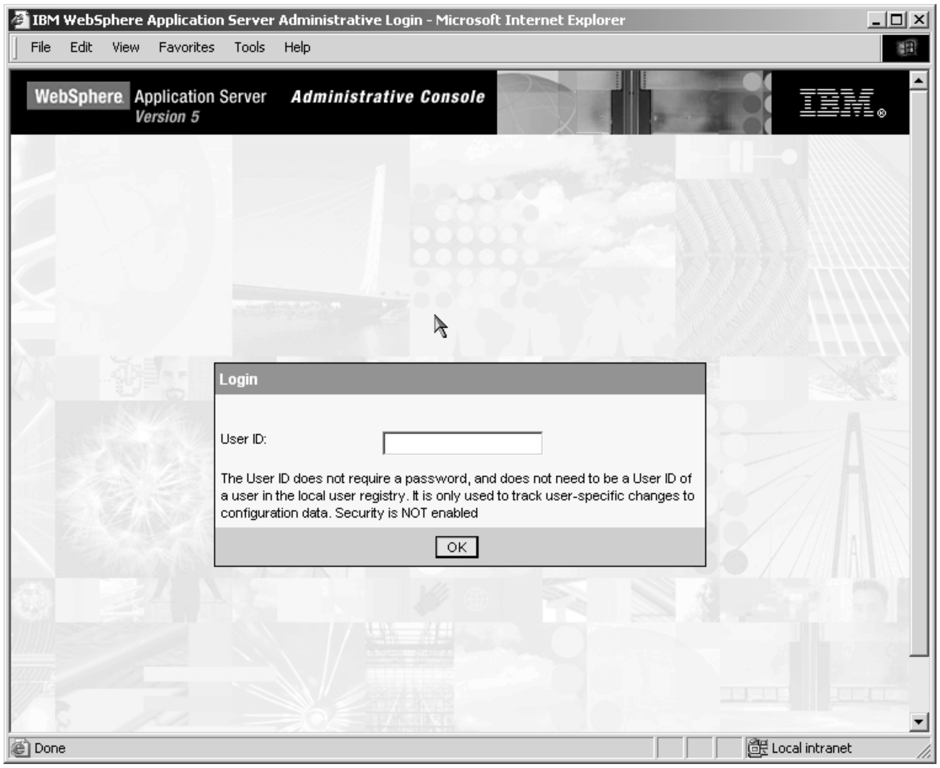
i. Click the **Tools** pull-down menu, and choose **Internet Options**.
ii. Click the **Privacy** tab.
iii. Click **Advanced**. Select **Accept** to accept first-party and third-party cookies.
iv. Click **OK.**

__ 4. Open a Web browser window and point it to:

```
http://<serverName>:<port>/admin
```

where *<serverName>* is the name of the class iSeries and *<port>* is your team instance's port number (see Task 4:, "Verifying that the application server has started" on page 6).

__ 5. You should see a login prompt page as shown in Figure 1-2.


IBM WebSphere Application Server Administrative Login - Microsoft Internet Explorer
File Edit View Favorites Tools Help
WebSphere Application Server
Version 5
Administrative Console
IBM
OK
Done
Local intranet

Login

User ID:

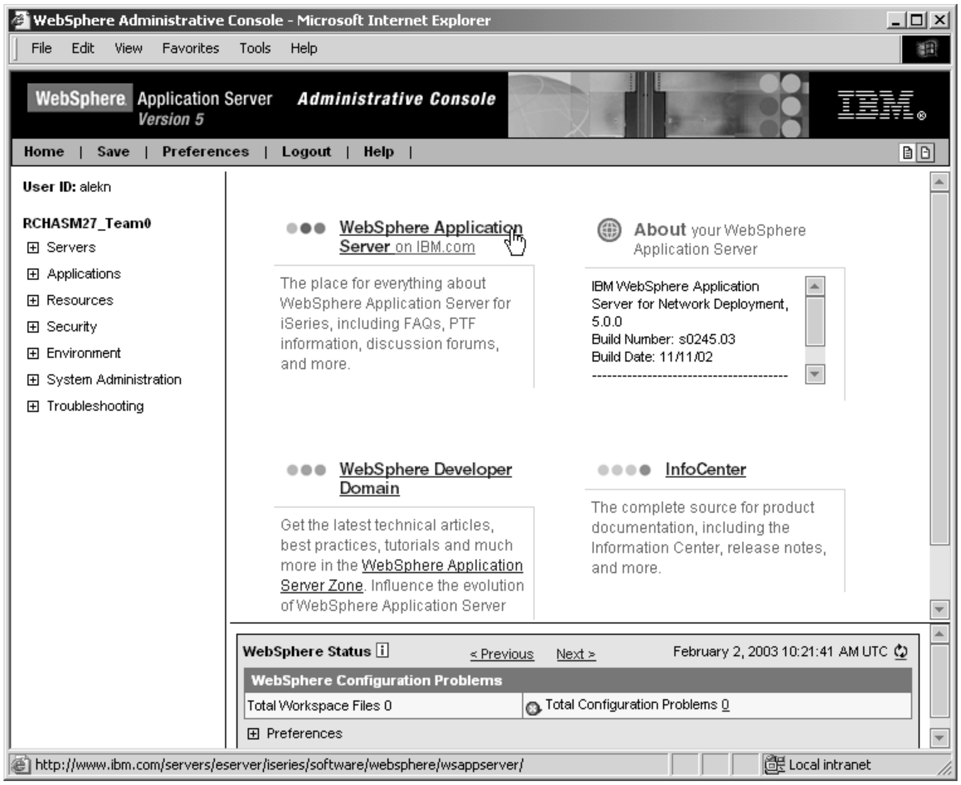
The User ID does not require a password, and does not need to be a User ID of a user in the local user registry. It is only used to track user-specific changes to configuration data. Security is NOT enabled

Figure 1-2 Login prompt page

__ 6. Enter your team user ID.

> **Note:** At this point, when the global security is not enabled, a user ID is used to track your admin console session's data.

__ 7. A welcome page is displayed (see Figure 1-3). On the left you should see a navigation tree. Click a plus sign icons and get familiar with the available options.


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
WebSphere Application Server Version 5 Administrative Console
IBM
Home | Save | Preferences | Logout | Help |
RCHASM27_Team0
⊞ Servers
⊞ Applications
⊞ Resources
⊞ Security
⊞ Environment
⊞ System Administration
⊞ Troubleshooting
WebSphere Application Server on IBM.com
About your WebSphere Application Server
WebSphere Developer Domain
InfoCenter
WebSphere Status
< Previous
Next >
WebSphere Configuration Problems
⊞ Preferences
Local intranet

User ID: alekn

The place for everything about WebSphere Application Server for iSeries, including FAQs, PTF information, discussion forums, and more.

IBM WebSphere Application Server for Network Deployment, 5.0.0
Build Number: s0245.03
Build Date: 11/11/02

Get the latest technical articles, best practices, tutorials and much more in the WebSphere Application Server Zone. Influence the evolution of WebSphere Application Server

The complete source for product documentation, including the Information Center, release notes, and more.

February 2, 2003 10:21:41 AM UTC

Total Workspace Files 0

Total Configuration Problems 0

http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/

Figure 1-3 The Administrative Console welcome page

__ 8. Click **Logout**.

__ 9. This concludes the lab.

Lab 2

Configuring an HTTP server instance

In this exercise you create a new instance of HTTP powered by Apache (V5.2) server instance. It is possible to use the internal HTTP server included with WebSphere Application Server to process requests for servlets and JavaServer Pages (JSPs). However in a production environment, a more robust Web server is required. You have the option to create a Web server instance on the same node as the application server, or you can establish a remote Web server to process requests for application server services. We configure an HTTP server instance on the same iSeries as the WebSphere Application Server.

Objectives

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

- Create a new HTTP powered by Apache (V5.2) server instance
- Associate the newly created HTTP server instance to an application server instance
- Start and use the HTTP server administration server
- Start and use the browser based Administrative Console
- Configure a virtual host
- Regenerate WebSphere plugin file
- Test your application server instance using verification tests

Time required

The time required to efficiently complete this lab project is 40 minutes.

Configuring an HTTP server instance

In this lab, you create a a new instance of IBM HTTP Server (powered by Apache) (V5R2). The HTTP server instance is not required to install WebSphere Application Server V5.0. However, we recommend that you use it to support requests for servlets and JSP resources managed by WebSphere Application Server.

To serve requests to the WebSphere Application Server instance created in the previous lab, we must associate the HTTP server with that WebSphere Application Server.

Task 1: Starting the *ADMIN instance of the HTTP server

To create an HTTP server configuration, you need to access the HTTP Administration server on the iSeries server. This is a special server instance that is capable of allowing administrators to modify server configuration and start, stop, and restart instances of the HTTP server remotely over the Web. The Administration server usually runs on port 2001. Most likely, the *ADMIN instance has been started. If not, perform the following steps:

__ 1. Start it from the command line:

i. Type the following command:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

ii. Press Enter.

__ 2. Or start it from iSeries Navigator:

i. Start iSeries Navigator.

ii. Double-click your iSeries server.

iii. Select **Network-> Servers-> TCP/IP**.

iv. Right-click **HTTP Administration** in the right frame.

v. Select **Start**.

Note: Do not select the Start Instance option from the menu. If the server is already stared, the Start option is disabled on the menu.

Task 2: Creating the HTTP Server configuration and instance (V5R2)

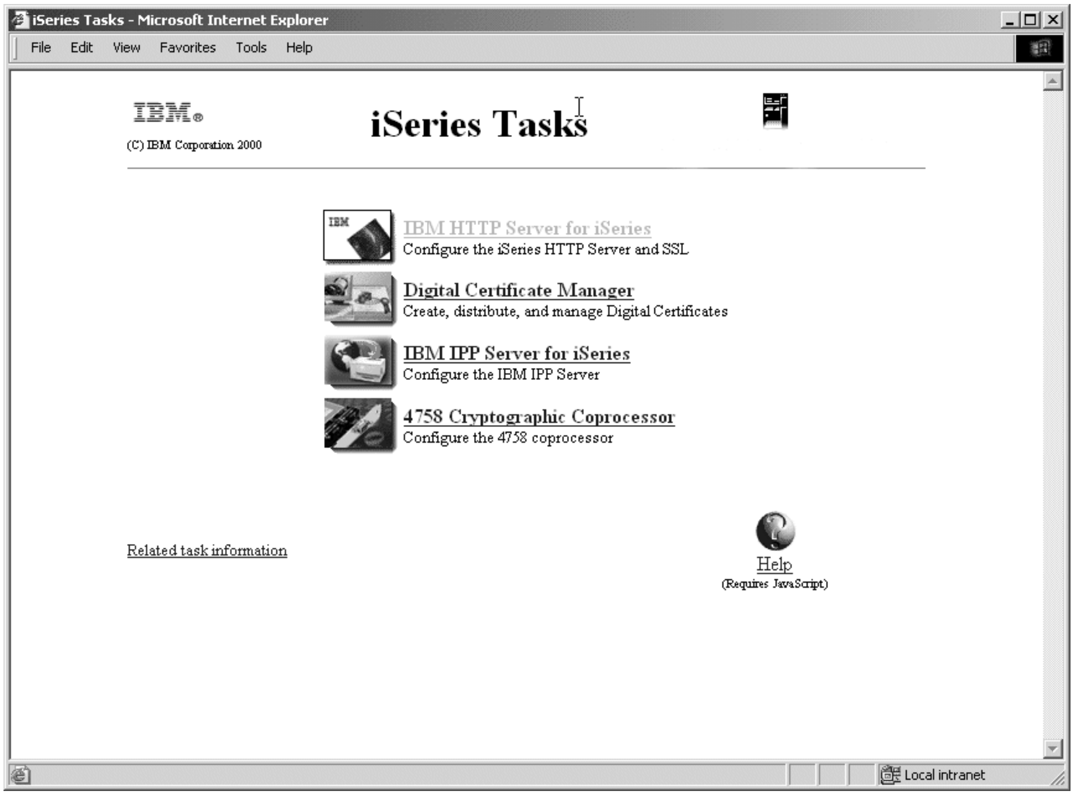
Add text here (Body0)

__ 1. Open the following URL in your browser:

```
http://MyAs400:2001
```

Here *MyAs400* is the host name of the classroom iSeries server.

__ 2. Enter your login/password information when prompted. You should see the page shown in Figure 2-1.


iSeries Tasks - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Local intranet

IBM®

(C) IBM Corporation 2000

iSeries Tasks

IBM HTTP Server for iSeries
Configure the iSeries HTTP Server and SSL

Digital Certificate Manager
Create, distribute, and manage Digital Certificates

IBM IPP Server for iSeries
Configure the IBM IPP Server

4758 Cryptographic Coprocessor
Configure the 4758 coprocessor

Related task information

Help
(Requires JavaScript)

Figure 2-1 iSeries Tasks page

__ 3. On the iSeries Tasks page, select **IBM HTTP Server for iSeries.**

__ 4. The HTTP Server for iSeries page is displayed. Click the **Setup** tab.

__ 5. Expand **Tasks and Wizards** in the left frame and click **Create New HTTP Server**.

__ 6. Select **HTTP server (powered by Apache)**, and click **Next**.

__ 7. In the right frame, type a name for the HTTP server instance. For this lab, use `TeamxHTTP` (where x is your team number). This name appears in the HTTPSVR subsystem when the job is running. Click **Next**.

__ 8. In the Create HTTP Server panel, select **No**, so the configuration is not based on an existing configuration. Click **Next**.

__ 9. Select the default for the server root name. Click **Next**.

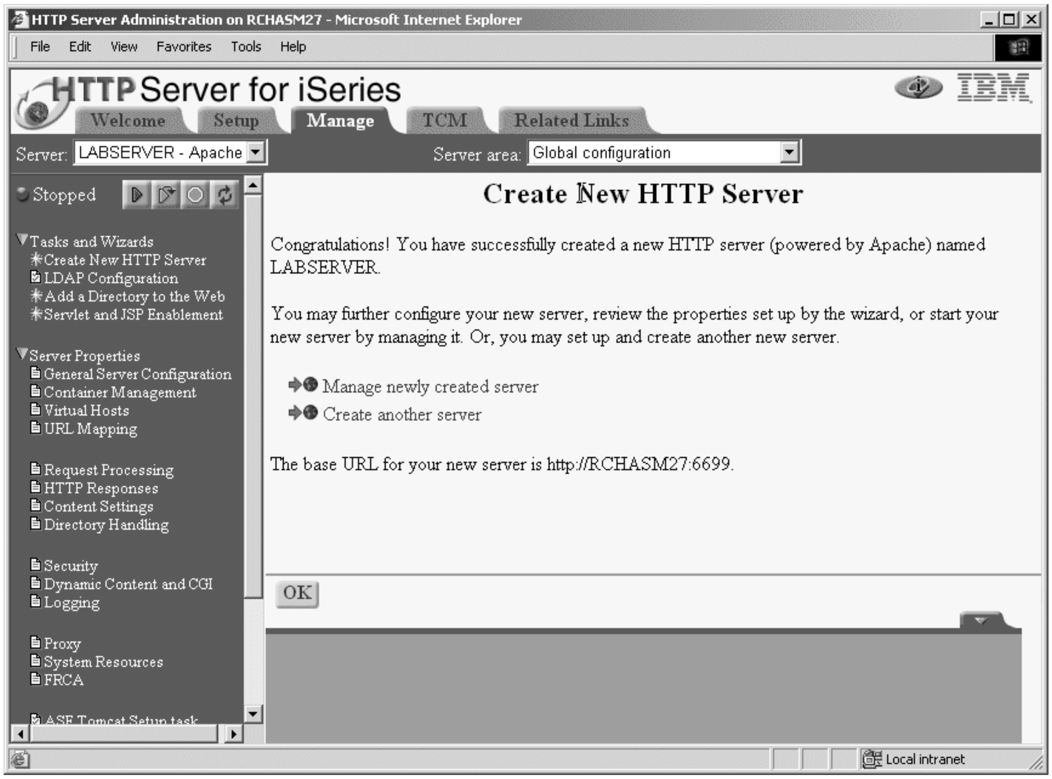
__ 10. In the next panel, set the document root (select the default). Click **Next**.

__ 11. Select the IP address and port. We accept requests from all IP addresses. Therefore that parameter can remain as is. The default for the port address, port 80, is different. The default server, **Server1**, created when the product was installed, has already assumed the port address of 80 for its HTTP server configuration. In addition, the other teams in the class require an unused port address for their HTTP server instance. The port that you will use is the port number you specified when you created your WebSphere Application Server instance in lab 1. That port number is already associated with your application server (it should be 6x00, where x is your team number). Enter your team's external HTTP port number.

__ 12. Select **Yes** for the logging option.

__ 13. Review your settings. To create the HTTP server instance, click **Finish**. To make changes, click **Back**.

__ 14. Now the configuration is created. The Congratulations page appears as shown in Figure 2-2.



Server: LABSERVER - Apache

Server area: Global configuration

Create New HTTP Server

Congratulations! You have successfully created a new HTTP server (powered by Apache) named LABSERVER.

You may further configure your new server, review the properties set up by the wizard, or start your new server by managing it. Or, you may set up and create another new server.

Manage newly created server

Create another server

The base URL for your new server is http://RCHASM27:6699.

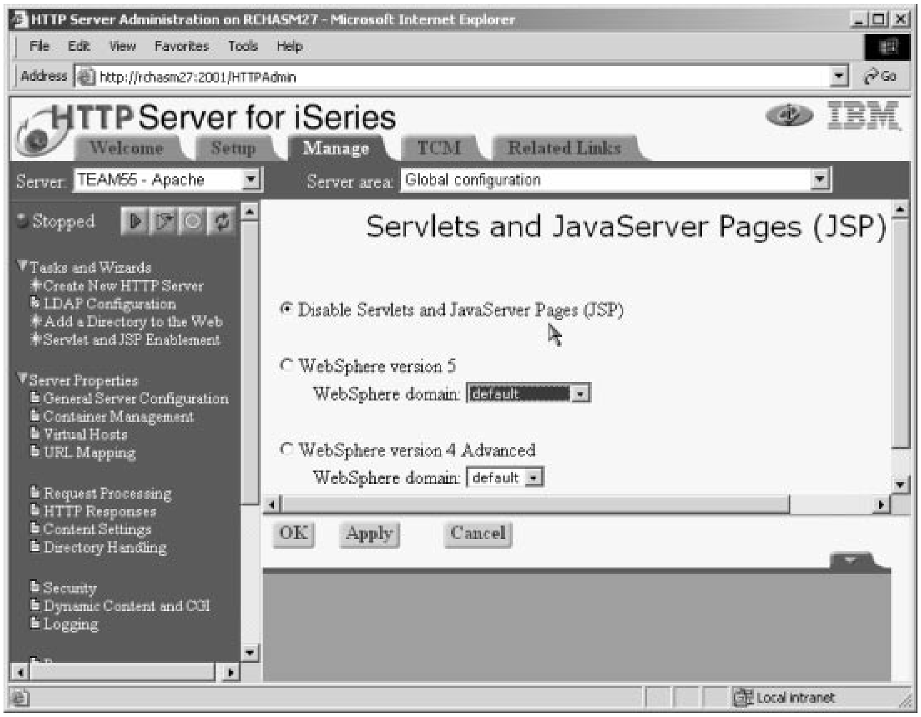
Figure 2-2 Congratulation page

__ 15. To enable your Web server to work with WebSphere Application Server, click **Manage newly created server**.

__ 16. With your new HTTP server selected, expand **Server Properties** in the left-hand frame.

__ 17. Click **WebSphere Application Server** (near the bottom of the frame).

__ 18. Select the version of WebSphere Application Server to configure, and select the WebSphere Application Server instance (just below using the drop-down arrow). See Figure 2-3.



Server: TEAM55 - Apache

Server area: Global configuration

Servlets and JavaServer Pages (JSP)

● Disable Servlets and JavaServer Pages (JSP)

○ WebSphere version 5
WebSphere domain: default

○ WebSphere version 4 Advanced
WebSphere domain: default

Figure 2-3 Configuring WebSphere plug-in

__ 19. To write the changes to the configuration file and return to the main configuration page, click **OK**.

Task 3: Starting the WebSphere Application Server environment

This task is split in two parts:

- Verifying that the WAS environment has been started
- Starting the administrative console

Verifying that the WebSphere Application Server environment is started.

__ 1. Run the Work with Active Jobs (WRKACTJOB) command, specifying the QEJBAS5 subsystem on the subsystem parameter:

```
WRKACTJOB SBS(QEJBAS5)
```

__ 2. Find your application server job. This should **Team<u>x</u>Svr** (where x is your team number). The application server job for the default WebSphere Application Server instance is named SERVER1. Your server job should be beneath this one.

__ 3. Specify option `5` (Work with Job) on the option line next to the job, press Enter.

__ 4. On the command line of the Work with Job display, specify option `10` (Display job log, if active), and press Enter.

__ 5. Press F10 to display all messages. Look for the message:

```
WebSphere application server application_server ready.
```

If the message is not displayed, press F5 to refresh log until the message is displayed.

When the message is displayed, the WebSphere Application Server environment has successfully started. It may take a few minutes to be displayed.

__ 6. To display the port number the application server is listening for the administration console, position the cursor on the last line of the message and press F1. This message should be displayed:

> WebSphere application server *application_server* in job *app_server_job* is ready to handle administration requests on port *port_number.*

__ 7. Press F3 twice to exit.

Starting the administrative console.

__ 1. To start the administrative console on the workstation, perform these steps:

i. Open this URL in your browser:

```
http://your.server.name:port/admin
```

Here *your.server.name* is the host name of the iSeries server on which the application server is running, and the *port* is the administration console port as noted in the message in the job log of the application server.

> **Note:** For the default WebSphere Application Server instance, the administration console port is 9090.

ii. When prompted, enter your user ID.

Task 4: Configuring the virtual host

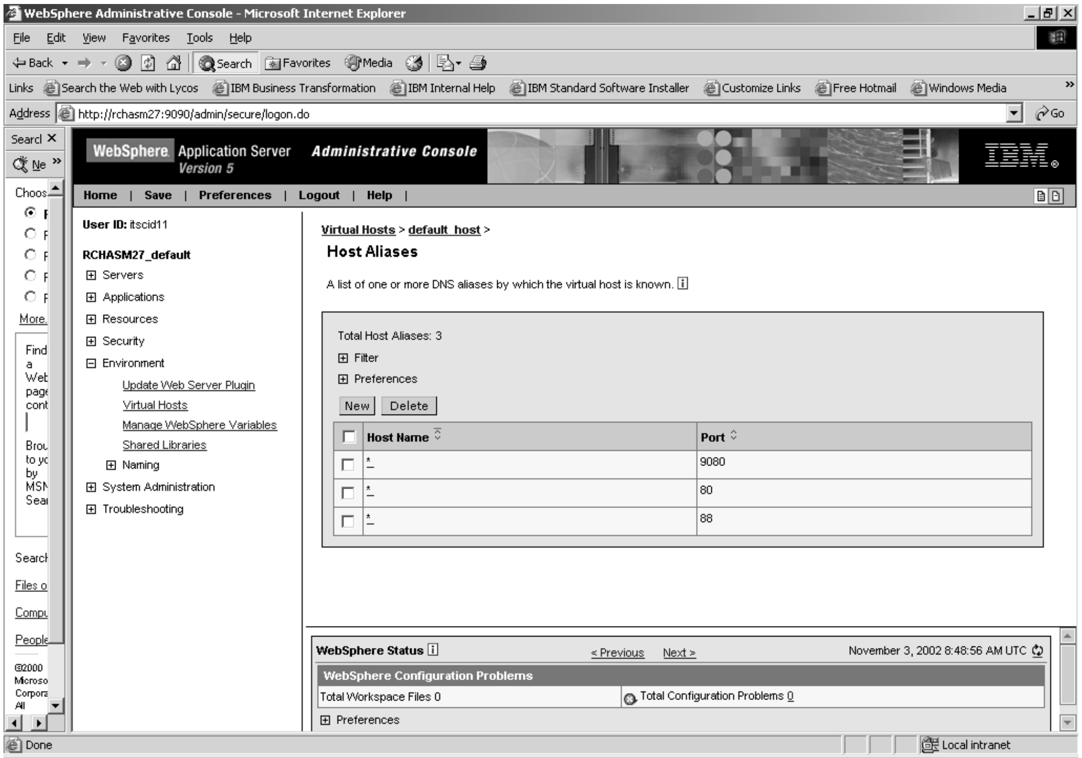
A “virtual host” is a configuration entity that allows WebSphere Application Server to treat multiple host machines or port numbers as a single logical host (virtual host) for configuration purposes. Each virtual host can be associated with multiple aliases. Each alias is a particular host name and port number. Combining multiple host machines into a single virtual host allows you to separate and control which WebSphere Application Server resources are available for client requests.

To update the Host Aliases table for the default_host virtual host from the administrative console, follow these steps:

__ 1. Expand **Environment-> Virtual Hosts** in the left frame of the administrative console.

__ 2. Click **default_host** in the right frame.

__ 3. Click **Host Aliases**. The settings for default_host are displayed (similar to Figure 2-4).


WebSphere Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help IBM Standard Software Installer Customize Links Free Hotmail Windows Media
Address http://rchasm27:9090/admin/secure/logon.do
Go
WebSphere Application Server Version 5 Administrative Console
IBM
Home | Save | Preferences | Logout | Help |
User ID: itscid11
RCHASM27_default
Servers
Applications
Resources
Security
Environment
Update Web Server Plugin
Virtual Hosts
Manage WebSphere Variables
Shared Libraries
Naming
System Administration
Troubleshooting
Filter
Preferences
New
Delete
WebSphere Status
< Previous
Next >
Done
Local intranet

Virtual Hosts > default_host >

Host Aliases

A list of one or more DNS aliases by which the virtual host is known.

Total Host Aliases: 3

☐	Host Name	Port
☐	*	9080
☐	*	80
☐	*	88

November 3, 2002 8:48:56 AM UTC

WebSphere Configuration Problems

Total Workspace Files 0	Total Configuration Problems 0

Figure 2-4 Host aliases page

__ 4. Verify that the host aliases table has the correct values for your instance: `6x10` and `6x00`, where x is your team number.

Note: The host names are set to "*" when you install WebSphere Application Server.

Note: If you update the host aliases table, regenerate the plug-in information. In the left pane, expand **Environment** and click **Update Web Server Plugin**. When the right frame appears with a prompt, click **OK**.

Starting IBM HTTP Server for iSeries

IBM HTTP Server for iSeries runs in the QHTTPSRV subsystem, and each HTTP server instance starts multiple jobs. The WebSphere Application Server code that plugs into IBM HTTP Server for iSeries runs in the HTTP server job that communicates with one or more application servers.

To start the HTTP server instance that you created, perform these steps:

__ 1. Open the following URL in your browser:

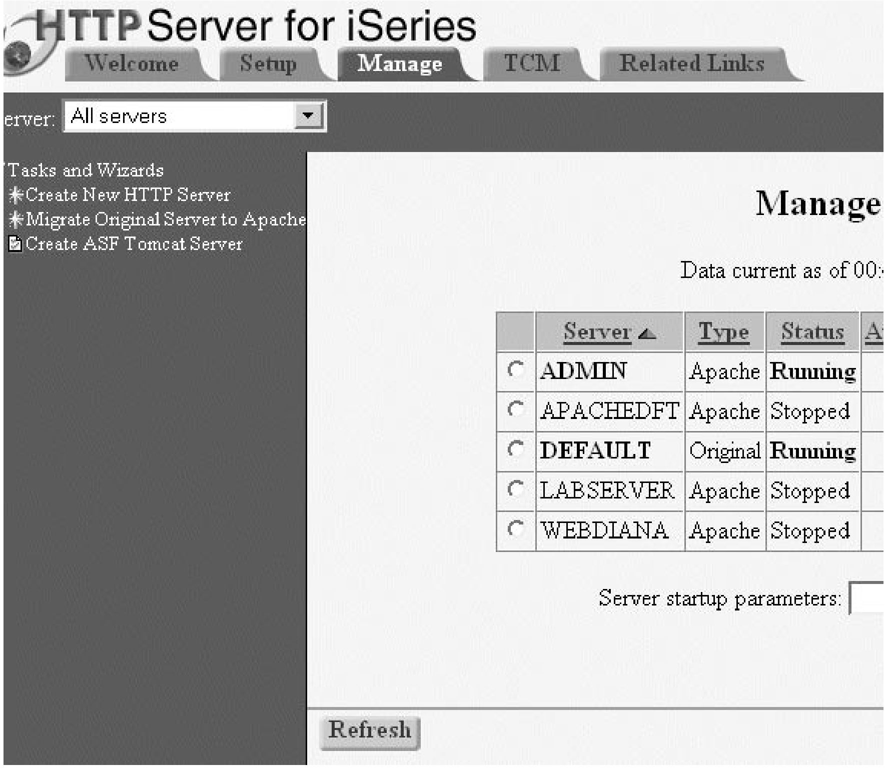
```
http://MyAs400:2001
```

Here *MyAs400* is the host name of the classroom iSeries server.

__ 2. Enter your login/password information when prompted.

__ 3. On the iSeries Tasks page, select **IBM HTTP Server for iSeries.**

__ 4. Select the **Manage** tab (see Figure 2-5).



Data current as of 00:

	Server ▲	Type	Status	A
○	ADMIN	Apache	Running	
○	APACHEDFT	Apache	Stopped	
○	DEFAULT	Original	Running	
○	LABSERVER	Apache	Stopped	
○	WEBDIANA	Apache	Stopped	

Server startup parameters:

Figure 2-5 Managing the HTTP server instances

__ 5. Make sure you have **All servers** selected in the **Server** pull-down box.

__ 6. Select the radio button by your server instance.

__ 7. Click **Start**.

Verifying the installation

You can verify your WebSphere Application Server installation by invoking sample browser-based applications or by invoking a verification script.

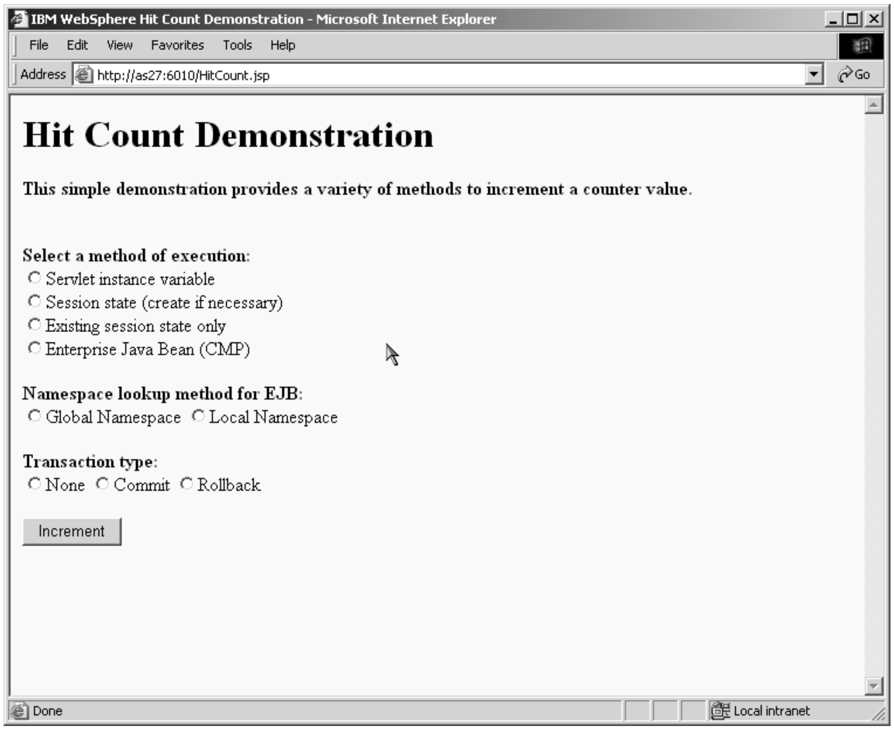
In this lab, we invoke the sample application:

__ 1. Open a Web browser window

__ 2. Point it to the HitCount application using the internal HTTP instance:

```
http://<serverName>:<port>/HitCount.jsp
```

Here *serverName* is the host name of the iSeries and *port* is the internal HTTP server port (it should be 6x10, where x is your team number). You should see the page shown at Figure 2-6).


IBM WebSphere Hit Count Demonstration - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
Go
Increment
Done
Local intranet

Address: http://as27:6010/HitCount.jsp

Hit Count Demonstration

This simple demonstration provides a variety of methods to increment a counter value.

Select a method of execution:

○ Servlet instance variable
○ Session state (create if necessary)
○ Existing session state only
○ Enterprise Java Bean (CMP)

Namespace lookup method for EJB:

○ Global Namespace ○ Local Namespace

Transaction type:

○ None ○ Commit ○ Rollback

Figure 2-6 Screen capture of running the HitCount application

__ 3. Run the same test using the external HTTP server instance.

Lab 3

Installing a J2EE application

You will begin this exercise by starting your WebSphere server. After the server has started, you will configure a JDBC Provider and a data source. The data source will be used by a J2EE application which you will install using the Administrative console. During installation you can specify binding information for the application running on your WebSphere Server. The application which you will install is a small banking application. The application features Container-Managed Persistence (CMP) and Local Interfaces EJB. CMP and local interfaces are part of the J2EE 1.3 specification. After you have finished installation, you will test the application.

Objectives

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

- Start WAS instance
- Create a new JDBC provider
- Create a new data source
- Install a J2EE application
- Test an application

Time required

The time required to efficiently complete this lab project is 1 hour.

J2EE applications

Most of the time, when you install a new application, there are some additional steps, like setting up a database access. In this exercise we walk you through the entire process of installing a J2EE applications with all additional steps.

Task 1: Starting the application server instance

To start an instance, if it's not started, run the startServer script from the Qshell command line:

__ 1. On the OS/400 command line, run the STRQSH (Start Qshell) command.

__ 2. On the Qshell command line, use the `cd` command to change to the directory that contains the script, for example:

```
cd /QIBM/ProdData/WebAS5/Base/bin
```

__ 3. Run the startServer script:

```
startServer -instance instance serverName
```

Here *instance* is the name of the instance you want to start (Teamx) and *serverName* is the name of the application server (TeamxSvr).

Note: you can have more that one application server created for a particular WAS instance. startServer script allows you to selectively start an application server

For example for Team1:

```
startServer -instance Team1 Team1Svr
```

Task 2: Verifying that the application server has started

Before you start the administrative console (which we do in the next lab exercise), verify that the environment started successfully. When the WebSphere Application Server environment is ready for use, a message is written to the job log of the application server job indicating that the WebSphere Application Server environment is ready.

To determine if the WebSphere Application Server environment is ready, perform these steps from an OS/400 command line:

__ 1. Run the Work with Active Jobs (WRKACTJOB) command, specifying the appropriate subsystem on the subsystem (SBS) parameter. For WebSphere Application Server (Base), use the QEJBAS5 subsystem:

```
WRKACTJOB SBS(QEJBAS5)
```

__ 2. Find your application server job (*TeamxSvr*, where *x* is your team number).

__ 3. Specify option `5` (Work with Job) on the option line next to the job.

__ 4. Press Enter.

__ 5. On the command line, specify option `10` (Display job log).

__ 6. Press Enter.

__ 7. Press F10

__ 8. Look for this message:

```
WebSphere application server application_server ready.
```

Here *application_server* is the name of your application server.

If the message is not displayed, press F5 to refresh the job log messages until the message is displayed. When the message is displayed, the WebSphere Application Server environment has successfully started. It may take up to 20 minutes for the message to be displayed, depending on your iSeries server. If the message is not displayed, see your instructor.

__ 9. To display the port number on which the application server is listening for the administrative console, position the cursor on the last line of the message.

__ 10. Press F1. This message is displayed:

```
WebSphere application server application_server in job app_server_job is ready to
handle administrative requests on port port_number
```

Here *application_server* is the name of your application server, *app_server_job* is the OS/400 job name for your application server, and *port_number* is the number of the port used by the administrative console.

__ 11. Press F3 twice to exit.

Task 3: Configuring a JDBC provider

__ 1. Start your administrative console for your instance by pointing your Web browser to:

```
http://<hostName>:port/admin
```

where <hostName> is the host name of the class iSeries system. *port* is the port number on which administrative console is listening.

__ 2. JDBC Providers, Data sources and other resources are organized under the **Resources** heading in the topology tree. Expand **Resources** and click **JDBC Providers** to configure the data source which your application will use to persist data (see Figure 3-1).

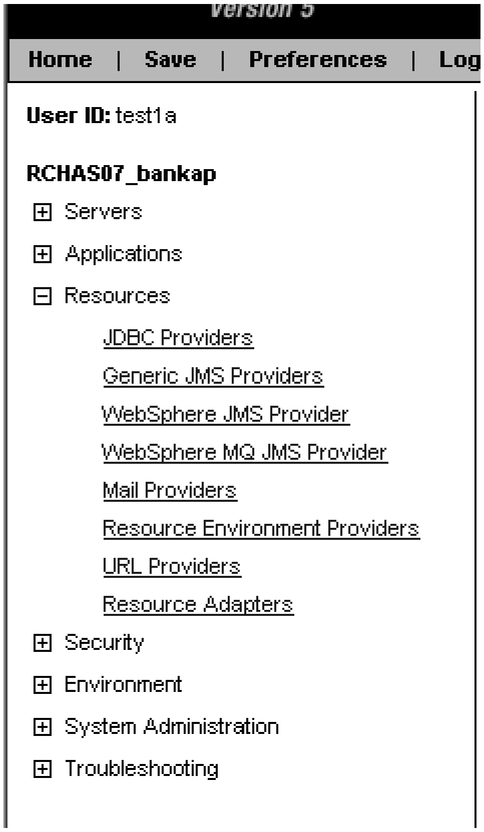
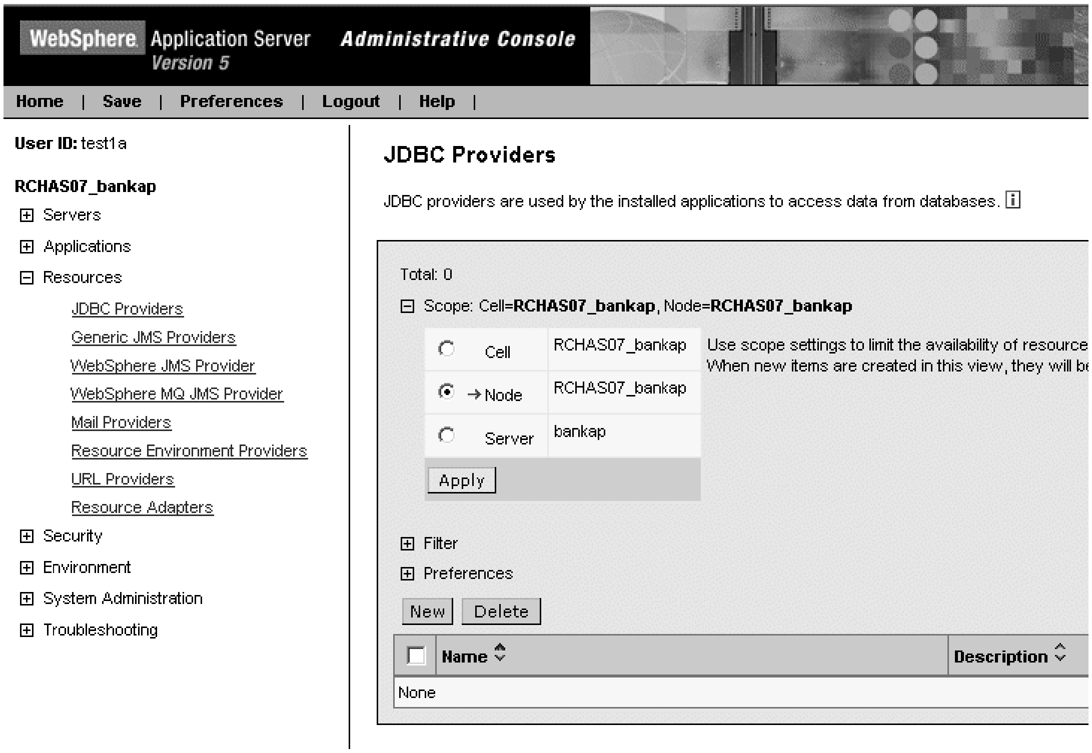


Figure 3-1 Selecting JDBC provider in the topology tree

__ 3. The existing JDBC Providers will be displayed. Verify that the Scope is set to Node, and then click **New** to create a DB2 Provider (see Figure 3-2).



JDBC Providers

JDBC providers are used by the installed applications to access data from databases.

Total: 0

⊟ Scope: Cell=**RCHAS07_bankap**, Node=**RCHAS07_bankap**

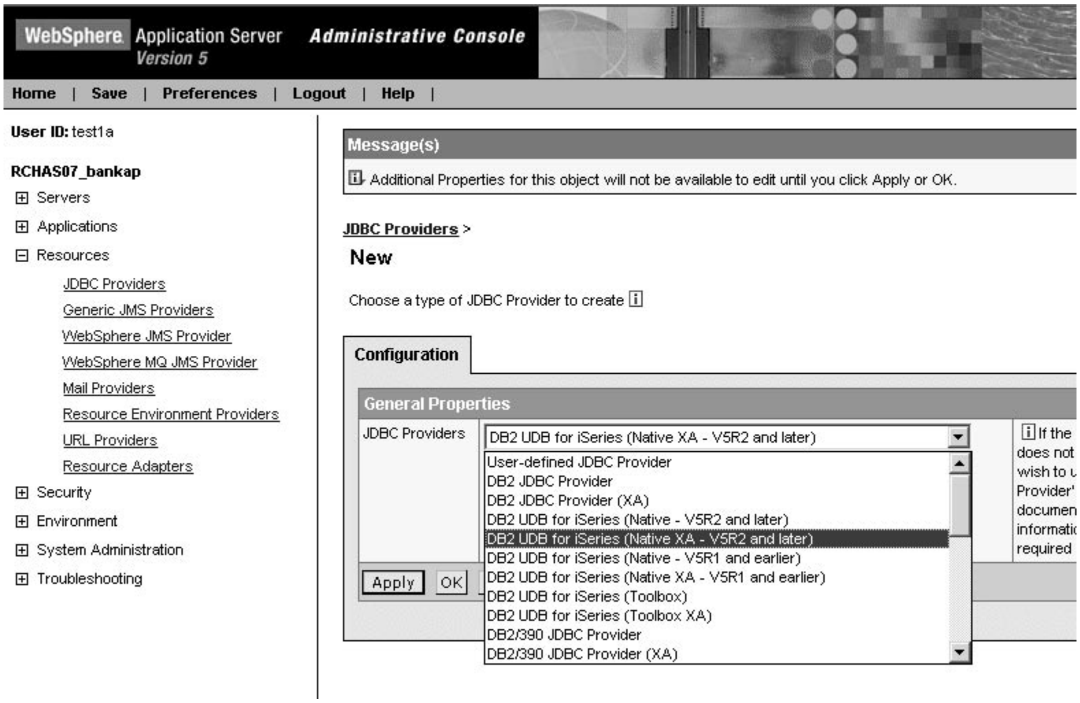
○ Cell	RCHAS07_bankap	Use scope settings to limit the availability of resource When new items are created in this view, they will be
● → Node	RCHAS07_bankap	
○ Server	bankap	

□	Name	Description
None		

Figure 3-2   Creating a new JDBC provider

__ 4. Select the DB2 JDBC Provider. For our MyBankCMP sample application we choose the **DB2 UDB for iSeries (Native XA - V5R2 and later)** driver.

__ 5. Press **Apply** (see Figure 3-3).



Message(s)

Additional Properties for this object will not be available to edit until you click Apply or OK.

JDBC Providers >

New

Choose a type of JDBC Provider to create

Configuration

General Properties

JDBC Providers: DB2 UDB for iSeries (Native XA - V5R2 and later)

- User-defined JDBC Provider
- DB2 JDBC Provider
- DB2 JDBC Provider (XA)
- DB2 UDB for iSeries (Native - V5R2 and later)
- DB2 UDB for iSeries (Native XA - V5R2 and later)
- DB2 UDB for iSeries (Native - V5R1 and earlier)
- DB2 UDB for iSeries (Native XA - V5R1 and earlier)
- DB2 UDB for iSeries (Toolbox)
- DB2 UDB for iSeries (Toolbox XA)
- DB2/390 JDBC Provider
- DB2/390 JDBC Provider (XA)

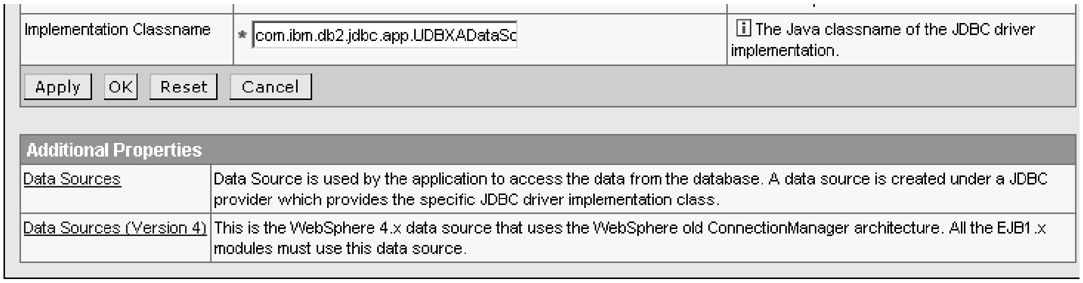
If the does not wish to u Provider' documen informati required

Figure 3-3   Selecting a JDBC driver

__ 6. The configuration properties for the JDBC Provider will be displayed. They are all correct, so no changes have to be done here. Click **OK**.

__ 7. You will now create a data source. Click the JDBC provider's name that you have just created. In our case **DB2 UDB for iSeries (Native XA - V5R2 and later)**.

__ 8. The configuration properties for the JDBC provider will be displayed. Below the configuration properties you should see two links as shown in Figure 3-4.



Implementation Classname	* com.ibm.db2.jdbc.app.UDBXADataSc	The Java classname of the JDBC driver implementation.

Additional Properties	
Data Sources	Data Source is used by the application to access the data from the database. A data source is created under a JDBC provider which provides the specific JDBC driver implementation class.
Data Sources (Version 4)	This is the WebSphere 4.x data source that uses the WebSphere old ConnectionManager architecture. All the EJB1.x modules must use this data source.

Figure 3-4 Selecting the Data Source link

__ 9. Click **Data Sources**.

__ 10. You should see the Data Sources screen. Click **New** to create a new data source.

__ 11. Provide the following information for the data source properties (see Figure 3-5):

i. Name: `MyBankDS`

This is the name of the Data Source and you can use any name.

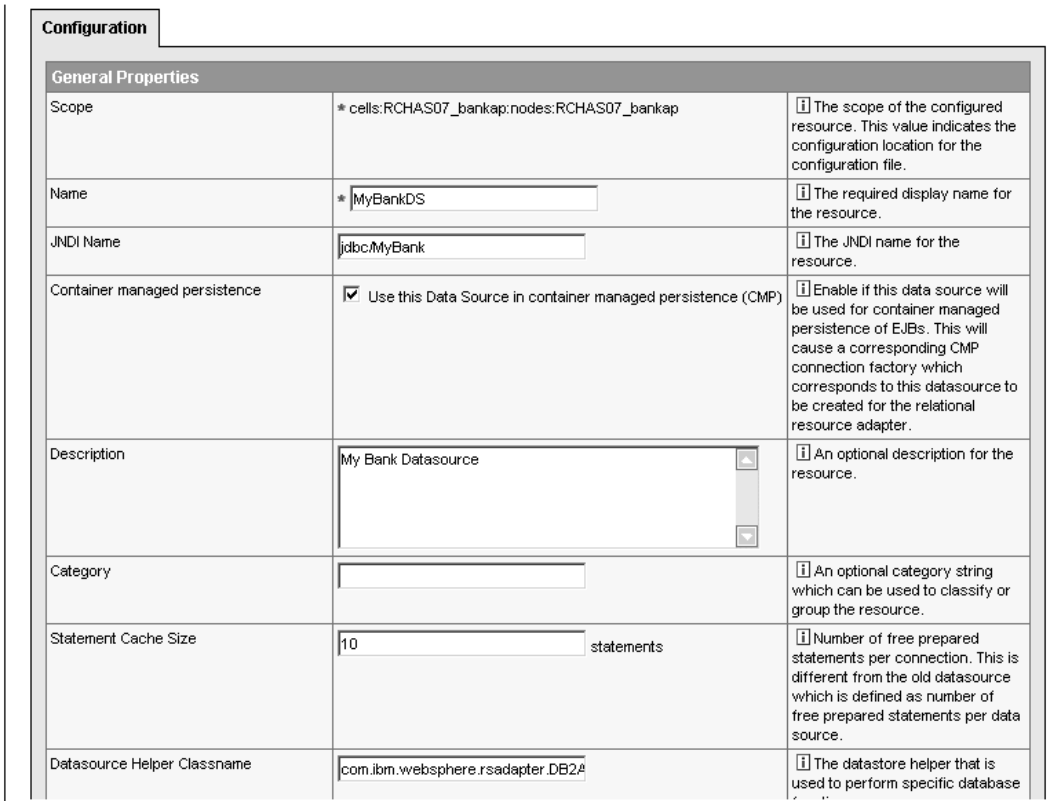
ii. JNDI name: `jdbc/MyBank`

iii. Select the checkbox next to **Use this DataSource in container managed persistence CMP**.

Selecting this checkbox will cause a corresponding CMP connection factory for the relational resource adapter to be created.

iv. Description: `MyBank Datasource`

__ 12. Click **OK** (you may have to scroll down).



General Properties

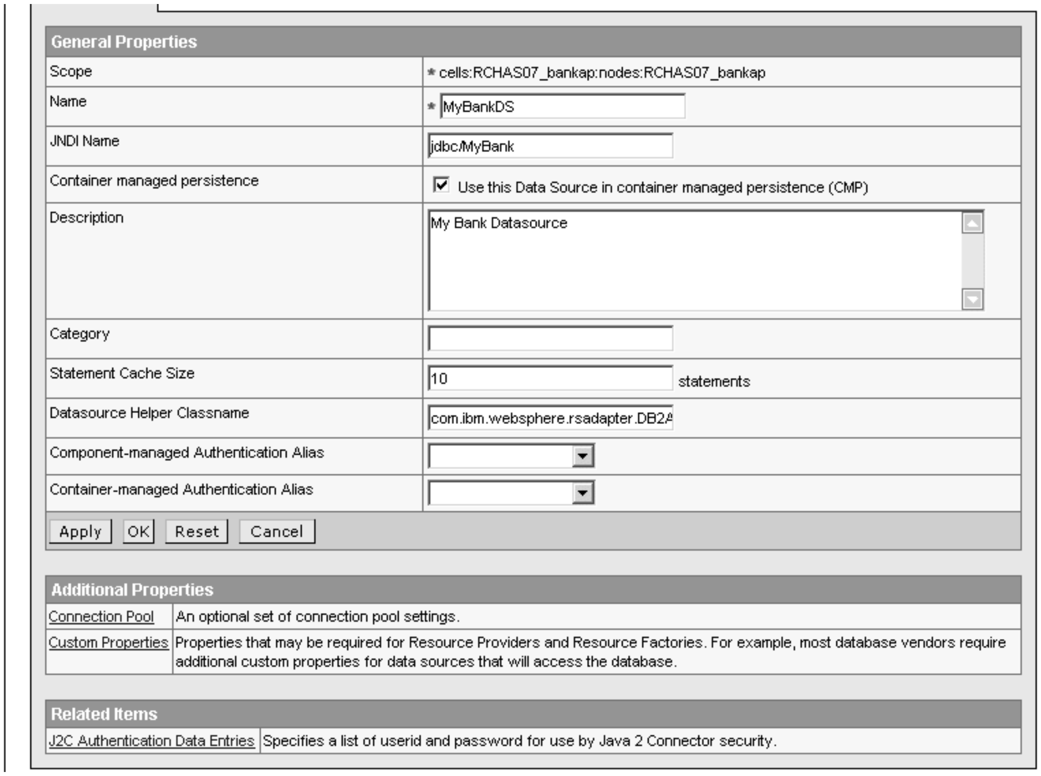
Scope	* cells:RCHAS07_bankap:nodes:RCHAS07_bankap	The scope of the configured resource. This value indicates the configuration location for the configuration file.
Name	* MyBankDS	The required display name for the resource.
JNDI Name	jdbc/MyBank	The JNDI name for the resource.
Container managed persistence	☑ Use this Data Source in container managed persistence (CMP)	Enable if this data source will be used for container managed persistence of EJBs. This will cause a corresponding CMP connection factory which corresponds to this datasource to be created for the relational resource adapter.
Description	My Bank Datasource	An optional description for the resource.
Category		An optional category string which can be used to classify or group the resource.
Statement Cache Size	10 statements	Number of free prepared statements per connection. This is different from the old datasource which is defined as number of free prepared statements per data source.
Datasource Helper Classname	com.ibm.websphere.rsadapter.DB2A	The datastore helper that is used to perform specific database

Figure 3-5 Entering a data source properties

Note: Ensure the **Use the Data Source in container managed persistence (CMP)** box is checked. If you forget to do this, you will have problems at runtime (the correspondent connection factory is not going to be created).

__ 13. The Data Source is now created, however there are a number of additional configuration properties which still need to be set. Click **MyBankDS**.

__ 14. The Data Source configuration properties will be listed again. Below the configuration properties, you should see several links. Click **J2C Authentication Data Entries** (you will have to scroll down, it is in the Related Items section). See Figure 3-6.



General Properties

Scope	* cells:RCHAS07_bankap:nodes:RCHAS07_bankap
Name	* MyBankDS
JNDI Name	jdbc/MyBank
Container managed persistence	☑ Use this Data Source in container managed persistence (CMP)
Description	My Bank Datasource
Category	
Statement Cache Size	10 statements
Datasource Helper Classname	com.ibm.websphere.rsadapter.DB2A
Component-managed Authentication Alias	
Container-managed Authentication Alias	

Additional Properties

Connection Pool	An optional set of connection pool settings.
Custom Properties	Properties that may be required for Resource Providers and Resource Factories. For example, most database vendors require additional custom properties for data sources that will access the database.

Related Items

J2C Authentication Data Entries	Specifies a list of userid and password for use by Java 2 Connector security.

Figure 3-6 Setting the authentication aliases

__ 15. The J2C Authentication Data Entries panel appears. Click **New** to create a new J2C Authentication Data Entry.

__ 16. Enter `MyBankAlias` as a name and your team's user ID and password.

Note: The user profile and password defined here must exist on your iSeries server. All connections obtained from this data source use user ID and password specified here.

__ 17. Click **OK**.

__ 18. Take a short cut back to the datasource: click **MyBankDS** in the link list located at the top of the frame.

__ 19. On the data source properties page scroll down and click **Custom Properties**.

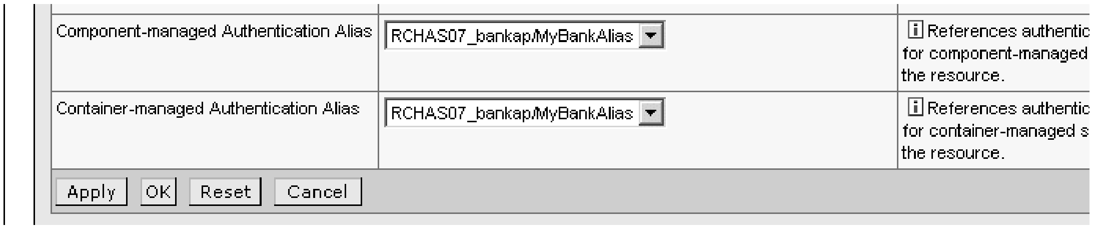
__ 20. Scroll down and click **databaseName** in the list of properties.

__ 21. Change this property from `*LOCAL` to `*LOCAL/Teamx`. Where Teamx is your team's collection.

__ 22. Click **OK**.

__ 23. Take the shortcut back to **MyBankDS**.

__ 24. On the data source properties page assign the above created J2C Authentication Data Entry to **Component-managed Authentication Alias** and also to **Container-managed Authentication Alias** (see Figure 3-7).



Component-managed Authentication Alias	RCHAS07_bankap/MyBankAlias	References authentic for component-managed the resource.
Container-managed Authentication Alias	RCHAS07_bankap/MyBankAlias	References authentic for container-managed s the resource.

Figure 3-7 Setting up authentication alias for data source

__ 25. Select **MyBankAlias** from the Component-managed Authentication Alias drop-down box.

__ 26. Select **MyBankAlias** from the Container-managed Authentication Alias drop-down box.

__ 27. Click **OK**.

__ 28. Your data source is now completely defined. Your configuration changes have to be saved. Click **Save** (either in the Message(s) box or on the menu bar).

__ 29. Click the **Save** button to confirm your changes.

Task 4: Installing a sample application

Use the administrative console for installing applications. To install the MyBankCMP application follow these steps:

__ 1. Be sure that your WAS instance is started and running.

__ 2. Start the administrative console for your instance.

__ 3. Expand **Applications** in the topology tree and click **Install New Application**.

__ 4. In the Preparing for the application installation panel select **Server path** radio button. This will allow you to select an application archive file from your iSeries. If the file is on the local workstation, you will have to select **Local path** radio button.

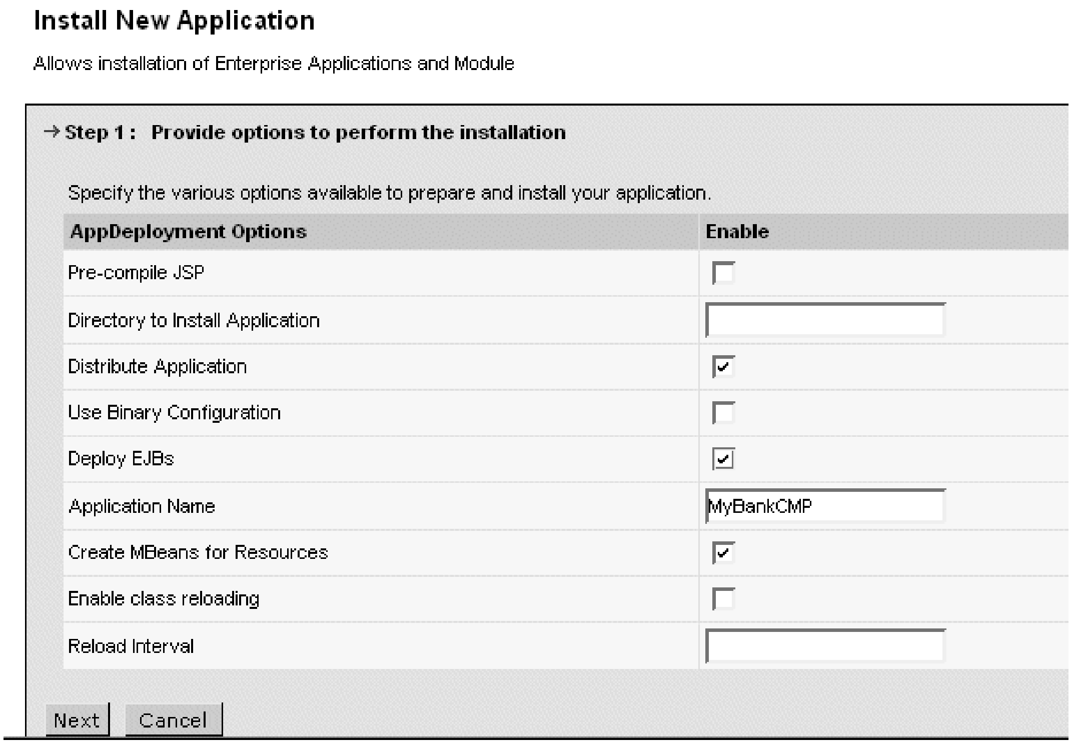
__ 5. Enter `/ITSO/mybankcmp.ear`

__ 6. Click **Next**.

__ 7. On next page, there are a number of settings which can be changed to customize your application. The only setting you will need to change is for the Connection Factory Bindings. Select the **Default connection factory bindings** radio button (you may have to scroll down) and enter **eis/jdbc/MyBank_CMP** for the JNDI name. Ensure **Per Connection Factory** is selected as **Resource authorization**.

__ 8. Click **Next**. The application installation wizard will initialize.

__ 9. Select the **Deploy EJBs** checkbox (see Figure 3-8).



Install New Application

Allows installation of Enterprise Applications and Module

→ Step 1 : Provide options to perform the installation

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

AppDeployment Options	Enable
Pre-compile JSP	□
Directory to Install Application	
Distribute Application	☑
Use Binary Configuration	□
Deploy EJBs	☑
Application Name	MyBankCMP
Create MBeans for Resources	☑
Enable class reloading	□
Reload Interval	

Figure 3-8 Installation wizard

__ 10. Click **Next**.

__ 11. The Step 2 panel has options to use when deploying enterprise beans. We'll accept the default values. Notice that the Database Type is DB2UDBISeries. Click **Next**.

__ 12. In the Step 3 panel you can specify the JNDI names for the Account and Transfer enterprise beans. Ensure Account and Transfer have been set to **ejb/MyBank/Account** and **ejb/MyBank/Transfer** respectively. Click **Next**.

__ 13. In the Step 4 panel you can specify a Data Source specifically for your EJB jar file:

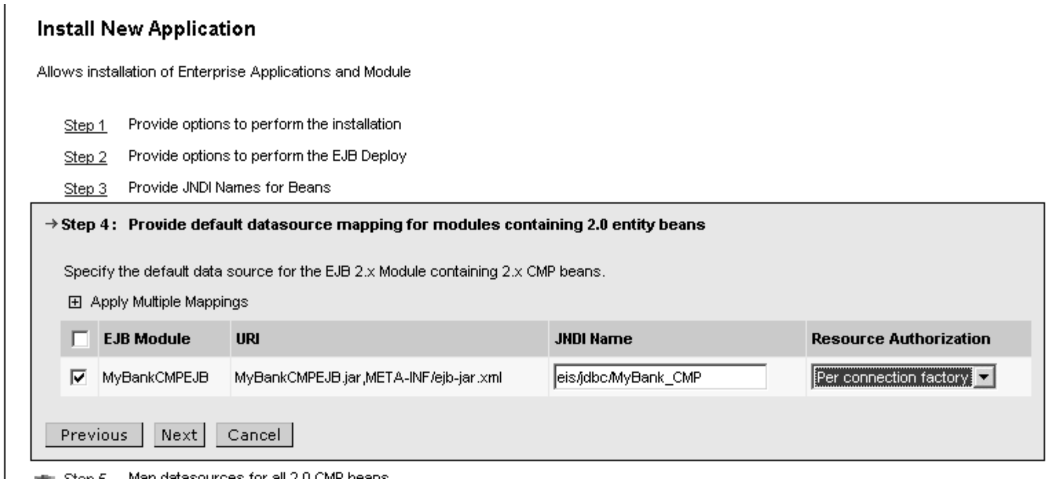
i. Expand **Apply Multiple Mappings**.

ii. Select your connection factory name from the first pull-down box.

iii. Click checkbox on the left of the **MyBankCMPEJB** module.

iv. Click the first Apply button underneath **Apply Multiple Mappings**. The connection factory name will be set for the EJB module.

v. Select **Per connection factory** in the **Resource Authorization** pull-down box (see Figure 3-9).



Install New Application

Allows installation of Enterprise Applications and Module

Step 1 Provide options to perform the installation

Step 2 Provide options to perform the EJB Deploy

Step 3 Provide JNDI Names for Beans

→ **Step 4 : Provide default datasource mapping for modules containing 2.0 entity beans**

Specify the default data source for the EJB 2.x Module containing 2.x CMP beans.

⊞ Apply Multiple Mappings

□	EJB Module	URI	JNDI Name	Resource Authorization
☑	MyBankCMPEJB	MyBankCMPEJB.jar,META-INF/ejb-jar.xml	eis/jdbc/MyBank_CMP	Per connection factory

Step 5 Map datasources for all 2.0 CMP beans

Figure 3-9 Mapping an EJB module to a datasource

__ 14. Click **Next**.

Note: Steps that require a user's input are marked with a red star on the left.

__ 15. In the Step 5 panel we can define a Data Source and a resource authorization for each entity EJB. We only have one entity EJB. Perform similar steps as in two previous steps. Click **Next**.

__ 16. In the Step 6 panel check the correctness of the JNDI names for the EJBs: **ejb/MyBank/Account** for the Account EJB and **ejb/MyBank/Transfer** for the Transfer EJB. Click **Next**.

__ 17. In the Step 7 panel you can select a virtual host for your application. We use the default virtual host. Click **Next**.

__ 18. In the Step 8 you can map your modules to the different application servers. In our case there is only one server, so no need to modify this information. Click **Next**.

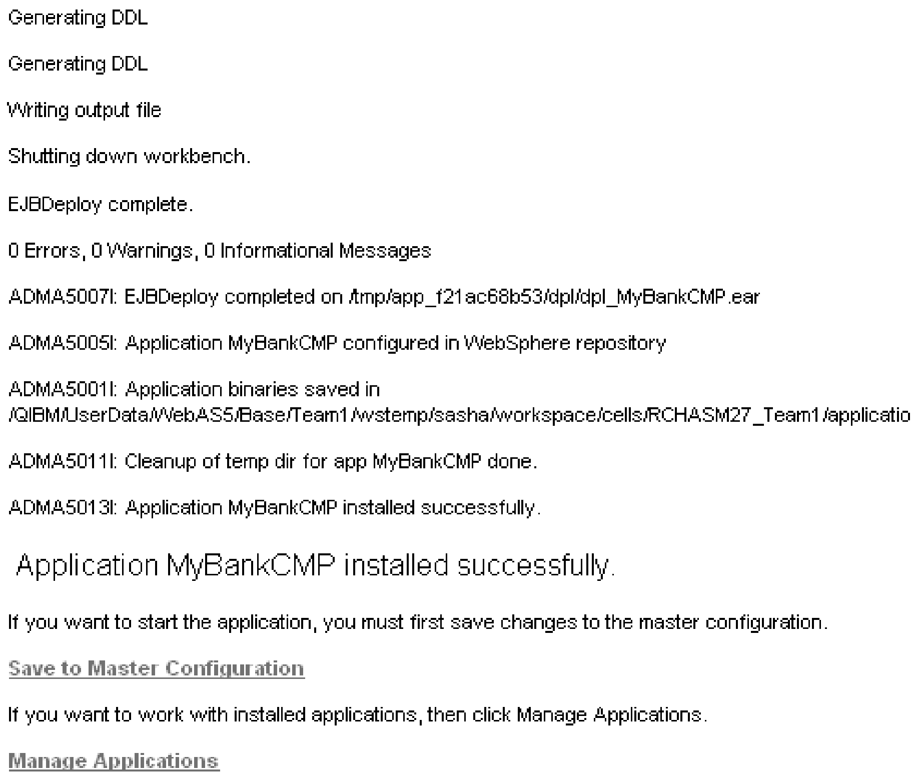
__ 19. In the Step 9 panel security can be defined and set for specific methods within your EJB module. We don't set any security information. Click **Next**.

__ 20. In the Step 10 panel you're presented with the summary of your configuration for the new application. Click **Finish**.

Note: You can skip any step(s) in the wizard by selecting a link directly to the desired step.

__ 21. WebSphere will go through the process of deploying your application. Figure 3-10 shows a sample messages that you can see in the administrative console during this process. This process may take a while.

__ 22. After application is successfully installed, save the changes by clicking **Save**.



Generating DDL

Generating DDL

Writing output file

Shutting down workbench.

EJBDeploy complete.

0 Errors, 0 Warnings, 0 Informational Messages

ADMA5007I: EJBDeploy completed on /tmp/app_f21ac68b53/dpl/dpl_MyBankCMP.ear

ADMA5005I: Application MyBankCMP configured in WebSphere repository

ADMA5001I: Application binaries saved in
/QIBM/UserData/WebAS5/Base/Team1/wstemp/sasha/workspace/cells/RCHASM27_Team1/applicatio

ADMA5011I: Cleanup of temp dir for app MyBankCMP done.

ADMA5013I: Application MyBankCMP installed successfully.

Application MyBankCMP installed successfully.

If you want to start the application, you must first save changes to the master configuration.

Save to Master Configuration

If you want to work with installed applications, then click Manage Applications.

Manage Applications

Figure 3-10 A sample screen during the installation of an application

Task 5: Testing the MyBankCMP application

Testing of the application includes 2 parts:

- Testing with the WebSphere internal HTTP server
- Testing with the external HTTP server which has been created in previous lab.

Go through the following steps:

__ 1. Expand **Applications** in the topology tree and select **Enterprise Application**.

__ 2. In the right pane click the checkbox on the left of **MyBankCMP**.

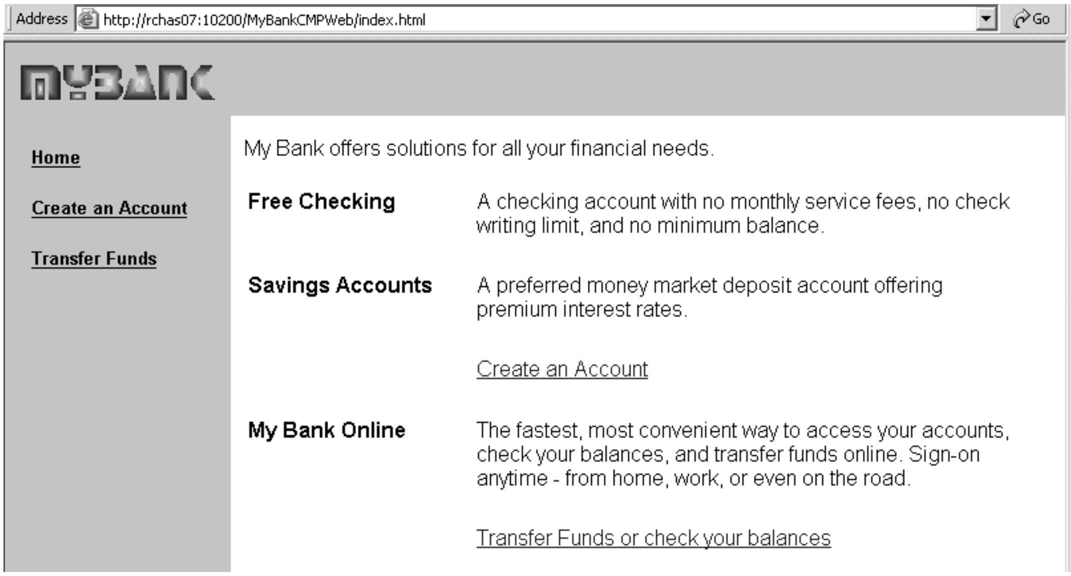
__ 3. Click **Start**. The application should start: icon in the Status column will change to the green arrow.

Note: If you move the cursor over the icon that represent the status of an application, a new pop-up box will display the meaning of each icon.

__ 4. Open a new Web browser window. Point it to the following URL (see Figure 3-11):

```
http://<hostName>:<port>/MyBankCMPWeb/index.html
```

where <hostName> is the name of the classroom iSeries. <port> is the number of the port on which the internal HTTP server for your instance is listening for incoming requests.



My Bank offers solutions for all your financial needs.

Free Checking A checking account with no monthly service fees, no check writing limit, and no minimum balance.

Savings Accounts A preferred money market deposit account offering premium interest rates.

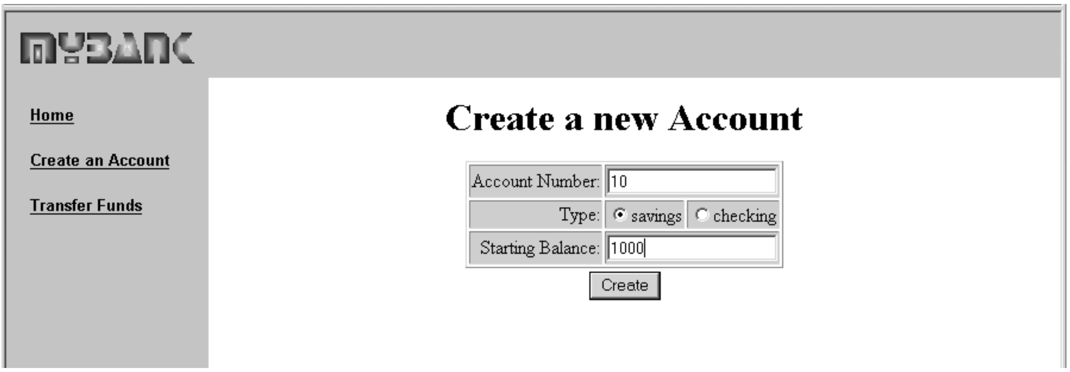
Create an Account

My Bank Online The fastest, most convenient way to access your accounts, check your balances, and transfer funds online. Sign-on anytime - from home, work, or even on the road.

Transfer Funds or check your balances

Figure 3-11 MyBankCMP welcome page

__ 5. Click the **Create an Account** link. Enter an Account Number, for example 10, and for the Starting Balance enter 1000, click **Create** (see Figure 3-12).



Create a new Account

Account Number: 10

Type: ● savings ○ checking

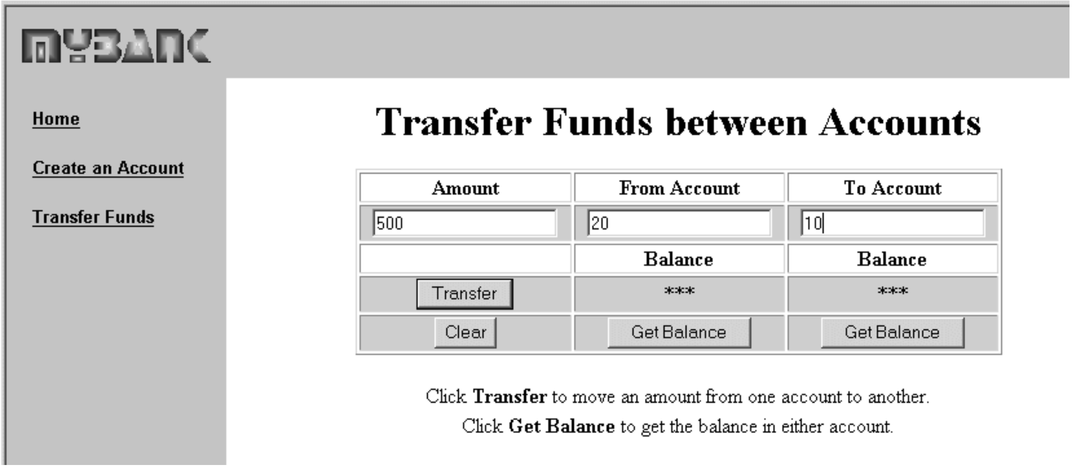
Starting Balance: 1000

Figure 3-12 Creating a new account

__ 6. Create a second account, use an Account Number of 20 and a Starting Balance of 4000. By now you've create 2 accounts. Now we want to test the Transfer Funds function.

__ 7. Click the **Transfer Funds** link.

__ 8. Enter `500` for `Amount` and `20` for the `From Account` and `10` for the `To Account`. Click **Transfer** (see Figure 3-13).



Transfer Funds between Accounts

Amount	From Account	To Account
500	20	10
	Balance	Balance
Transfer	***	***
Clear	Get Balance	Get Balance

Click **Transfer** to move an amount from one account to another.
Click **Get Balance** to get the balance in either account.

Figure 3-13 Transferring funds

__ 9. Now move on. We need to test our application with the external HTTP server. In order to do that we need:

i. Regenerate the WebSphere plugin information

ii. Restart the HTTP server instance

__ 10. Expand **Environment** in the topology tree.

__ 11. Click **Update Web Server Plugin**.

__ 12. Click **OK** in the frame to the right of the topology tree. The plugin-cfg.xml file will be update. The HTTP server instance reads in information from this file during startup.

__ 13. Restart your HTTP server instance:

i. Open the following URL in your browser:

```
http://<MyAs400>:2001
```

Here *MyAs400* is the host name of the classroom iSeries server.

ii. Enter your login/password information when prompted.

iii. On the iSeries Tasks page, select **IBM HTTP Server for iSeries.**

iv. Select the **Manage** tab.

v. Make sure you have **All servers** selected in the **Server** pull-down box.

vi. Select a radio button to the left of your server instance name.

vii. Click **Stop**.

viii.Refresh screen by clicking **Refresh**.

ix. When server is stopped, click **Start** to restart the HTTP server instance.

__ 14. Test the application again. This time use the external HTTP server port:

```
http://<hostName>:<extHTTPport>/MyBankCMPWeb/index.html
```

where <hostName> is the name of the classroom iSeries. <extHTTPport> is the number of the port on which the external HTTP server for your instance is listening for incoming requests. It should be 6x00.

__ 15. This concludes the lab.

Lab 4

WebSphere administrative tools: wsadmin

In this exercise, you change WAS configuration using wsadmin tool.

Objectives

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

- Start wsadmin interactively
- Modify WAS configuration
- Write a simple Jacl script file
- Save configuration

Time required

The time required to efficiently complete this lab project is 40 minutes.

Introduction to wsadmin

WebSphere Application Server 5.0 includes a scripting solution called *wsadmin*. Wsadmin allows an administrator to configure and control a WebSphere installation, much as wscp is used to administer WebSphere Application Server 4.0 installations. But wsadmin has a very different feel to it than wscp, chiefly this is due to the use of the Bean Scripting Framework (hereafter, BSF), which allows a variety of scripting languages to be used by script writers. You may use any scripting language supported by BSF to write scripts that configure and control your WebSphere installation, but only the Jacl language has been tested and is supported in this initial release. Note that various places in this document refer to other languages, but it is still true that Jacl is the only supported language.

The general pattern for the operation of WebSphere scripts is this: wsadmin makes various Java objects available through language-specific interfaces, and scripts use these objects to communicate with MBeans running in WebSphere server processes. MBeans are JMX management objects: they define the management interface for various pieces of the runtime. Some of the objects made available by wsadmin implement configuration actions, others implement control actions, and still others provide convenience functions or on-line help. The syntax employed by scripts is determined by the scripting language engine that interprets the scripts, and not by WebSphere itself. The supported languages all have ways to invoke methods on the declared Java objects, and the process of writing WebSphere 5.0 scripts involves invoking these methods to accomplish some goal.

There are several sorts of server processes to which a scripting client can connect: application servers, node agents, and network deployment managers. Different actions are possible when connected to different types of processes; also, the MBeans visible to the scripting client vary depending on what sort of process is connected. When connected to a network deployment manager, all MBeans in all server processes are accessible to the client. When connected to a node manager, all MBeans in all server processes on that node are accessible. When connected to an application server, only the MBeans running in that application server are visible.

Task 1: Starting wsadmin

In order to run wsadmin interactively, you should start QShell. You can use wsadmin with WAS server started or stopped. However, the way you start is slightly different. Lets do some exercises to practice wsadmin:

__ 1. On the Qshell command line, use the `cd` command to change to the directory that contains the create instance script, for example:

```
cd /QIBM/ProdData/WebAS5/Base/bin
```

__ 2. Stop your instance, if it is not stopped, by executing the following command at QShell command line:

```
stopServer -instance Teamx TeamxSvr
```

__ 3. Start wsadmin in interactive mode:

```
wsadmin -instance Teamx -conntype NONE
```

Note: We use *conntype* parameter to eliminate the error message:

```
WASX7023E: Error creating "SOAP" connection to host "localhost";
```

This message is displayed because the server is not running and there is no SOAP port to connect to.

instance parameter points to the instance which we want to administer. If you skip this parameter, wsadmin is connected to the default instance.

Task 2: Exploring wsadmin objects

__ 1. There are four objects that you can use in scripts:

- AdminControl
- AdminConfig
- AdminApp
- Help

__ 2. We explore each object in some details. First, we use Help object. Run the following command at wsadmin command line:

```
$Help help
```

> **Note:** Always wait for the **wsadmin >** prompt to appear before executing the next command.
>
> Replace x in all sample commands.

__ 3. If display what commands are available with Help object. In general, you can use *help* command with all objects. It will provide general help about available commands with an object. For example, `$AdminConfig help`

__ 4. Use the AdminConfig object to invoke configuration commands and create or change elements of the WebSphere Application Server configuration. Run the following command to display the available options with AdminConfig object:

```
$AdminConfig help
```

__ 5. Now we try to execute one of the commands:

```
$AdminConfig showall
```

__ 6. wsadmin provides help information. It points that you have to provide config id parameter. Lets provide a config id for an application server. Run the following commands:

```
set svr [$AdminConfig getid /Server:TeamxSvr/]
$AdminConfig showall $svr
```

The first command sets a variable *svr* to the config id of the *TeamxSvr* server. The second command display configuration information for the TeamxSvr server. You can scroll the list of the attributes.

__ 7. Now lets try to change some attributes. Some of the attributes are nested deep into the configuration data. In order to get to them, we are going to traverse the configuration tree. We are going to change the Host Aliases table for the *default_host* virtual host:

i. Display the available virtual hosts:

```
$AdminConfig list VirtualHost
```

ii. You will see the list of 2 hosts. Next we display the attributes for default_host by running 2 commands:

```
set vh [$AdminConfig getid /VirtualHost:default_host/]
$AdminConfig showall $vh
```

The first command creates a variable with the virtual host's id. The second command displays the virtual host's attributes. Page up and look at the Host Aliases table information. For example:

```
> $AdminConfig showall $vh
   {aliases {{{hostname *}
   {port 6210}} {{hostname *}
   {port 6200}}}}
   {mimeTypes {{{extensions STL;stl}
   {type application/SLA}} {{extensions
```

iii. Now we will update the host aliases table (enter this command on one line; watch for the correct number of the braces):

```
$AdminConfig modify $vh {{aliases {{{hostname *} {port 6x51}} {{hostname *} {port
6x52}}}}}
```

where you replace x with your team's number.

Pay attention to the syntax. The simples way to get it right is to copy the attributes from the *showall* command, modify the values and use this information to update the configuration. Don't forget to surround the attributes with the pair of curly braces.

iv. Run *showall* command again. You should see your changes:

```
> $AdminConfig showall $vh
   {aliases {{{hostname *}
   {port 6210}} {{hostname *}
   {port 6200}} {{hostname *}
   {port 6251}} {{hostname *}
   {port 6252}}}}
```

v. However, your changes are not written to the master configuration. They are saved in `/QIBM/UserData/WebAS5/Base/<instanceName>/wstemp/Scriptf22a07f913`. *Scriptf22a07f913* is a random name and you should see a different folder name.

vi. You have to execute one more command to save your changes (wsadmin in interactive mode doesn't save your changes):

```
$AdminConfig save
```

__ 8. Lets look at another object - AdminControl. You can use the AdminControl object to invoke operational commands on objects that run in the WebSphere Application Server.

Examples of methods available for AdminControl include: *getAttribute*, *setAttribute*, and *invoke*. Many of the AdminControl methods have multiple signatures so that they can be invoked in a raw mode using parameters specified by Java Management Extensions (JMX), or using strings for parameters.

Note: Most users should use string signatures.

__ 9. Execute the following command:

```
$AdminControl getHost
```

__ 10. You should see the following error:

```
WASX7015E: Exception running command: "$AdminControl getHost"; exception
information:
 com.ibm.ws.scripting.ScriptingException: AdminControl service not available
```

__ 11. This says that the server has not been started and AdminControl object is not available.

__ 12. Exit from wsadmin interactive mode:

```
exit
```

__ 13. Start your instance:

```
startServer -instance Teamx TeamxSvr
```

__ 14. And when it is up, start wsadmin:

```
wsadmin -instance Teamx
```

__ 15. This time you don’t need to specify the *conntype* parameter. Try to execute getHost command for $AdminControl object:

```
$AdminControl getHost
```

__ 16. Now we are going to stop and start an application:

i. Run the following command (enter this command on one line):

```
set am [$AdminControl queryNames type=ApplicationManager,process=TeamxSvr,*]
```

This command creates a variable for the application manager in the *TeamxSvr* server.

ii. Stop the *ivtApp* application:

```
$AdminControl invoke $am stopApplication {"ivtApp"}
```

iii. Switch to the administrative console and verify that the correct status is shown.

iv. Start the application:

```
$AdminControl invoke $am startApplication {"ivtApp"}
```

__ 17. Lets look at AdminApp object. Use the AdminApp object to install, modify, and administer applications.

The AdminApp object interacts with the WebSphere Application Server management and configuration services in order to make application inquiries and changes. This includes installing and uninstalling applications, moving modules, listing modules, exporting, and so on. The AdminApp object is different from the AdminConfig and AdminControl objects because in order to use AdminConfig and AdminControl, for the most of the operations, the scripting process must be connected to a server. You can invoke most of the AdminApp functions in local mode, which means that the client does not communicate with the server to accomplish the function.

__ 18. To practice with AdminApp object, look at the help:

```
$AdminApp help
```

__ 19. As an exercise export the application:

i. List the applications:

```
$AdminApp list
```

ii. Export one of them, for example ivtApp:

```
$AdminApp export "ivtApp" /tmp/ivtAppTeamx.ear
```

iii. In Windows Explorer navigate to the <X>:\tmp (<X> is a drive mapped to the *root* share of your iSeries system) directory on your iSeries system and verify that the EAR file has been exported.

Task 3: Install a J2EE application

You can use wsadmin tool to install and uninstall an application:

__ 1. We will use interactive mode for installing an application. The application EAR file will be shared between all teams:

i. Run the following command:

```
$AdminApp installInteractive /ITSO/HelloWorld.ear
```

ii. wsadmin will ask you several questions about installing options. It will provide the default values in the brackets. For example, the first questions will look like:

```
Web Module:  HelloWorldWeb
URI:  HelloWorldWeb.war,WEB-INF/web.xml
Virtual Host:  [default_host]:
```

iii. In order to select the default option, just hit Enter.

iv. The second questions is about the server (where to install a Web module). Hit Enter accepting a default value.

v. Next question propose to precompile JSPs. We accept default value [No] because there is no JSP in our application. Hit Enter.

vi. Accept default for directory to install the application.

vii. Use the default values for until you are asked if you want to enable class reloading. Enter `Yes` and hit Enter.

viii.Set reload interval to `600` (sec.).

ix. Save the configuration:

```
$AdminConfig save
```

x. List all application to verify the installation

```
$AdminApp list
```

You should see HelloWorld in the list.

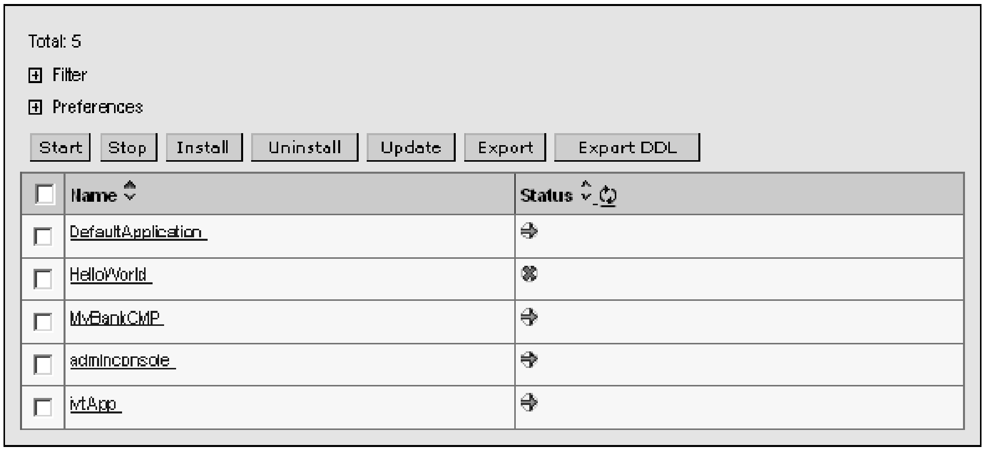
xi. Start the administrative console by pointing your Web browser to the following URL:

```
http://<hostName>:<port>/admin
```

__ 2. Expand **Applications**.

__ 3. Click **Enterprise Applications**.

You should see the HelloWorld application in the list (see Figure 4-1).



□	Name	Status
□	DefaultApplication	➜
□	HelloWorld	✖
□	MyBankCMP	➜
□	adminconsole	➜
□	ivtApp	➜

Figure 4-1 Installed applications

__ 4. Start the application. You can do it either from wsadmin command line or from the administrative console interface. If you use wsadmin, this is the command:

```
$AdminControl invoke $am startApplication {"HelloWorld"}
```

__ 5. Open a new Web browser window and access the following URL:

```
http://<hostName>:<port>/HelloWorldWeb/HelloWorldServlet
```

where <port> is the internal port for your instance. If you want to test the application with the external port, regenerate the plug-in file and restart the HTTP server.

__ 6. You should see **Hello World!**

Lab 5

Enabling Global Security (optional)

In this exercise you will secure the WebSphere v5.0 Administrative console and then verify that access is limited to certain functions.

Objectives

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

- Enable Global Security and verify it

Time required

The time required to efficiently complete this lab project is 30 minutes.

Global security

Global security applies to all applications running in the environment and determines whether security is used at all, the type of registry against which authentication takes place, and other values, many of which act as defaults.

The term global security implies the security configuration which is effective for the entire security domain. A security domain consists of all servers configured with the same user registry realm name. In some cases, the realm can be the machine name of a LocalOS user registry. In this case, all application servers must reside on the same physical machine. In other cases, the realm can be the machine name of an LDAP user registry. Since LDAP is a distributed user registry, a multiple node configuration is supported, such as the case for a Network Deployment environment. The basic requirement for a security domain is that the access ID returned by the registry from one server within the security domain is the same access ID as that returned from the registry on any other server within the same security domain. The access ID is the unique identification of a user and is used during authorization to determine if access is permitted to the resource.

Configuration of global security for a security domain consists of configuring the common user registry, the authentication mechanism, and other security information, which defines the behavior of a security domain. The other security information which you can configure includes Java 2 Security Manager, Java Authentication and Authorization Service (JAAS), Java 2 Connector authentication data entries, CSIv2 or SAS authentication protocol (RMI over IIOP security), and other miscellaneous attributes. The global security configuration usually applies to every server within the security domain.

Administrative roles in WAS

WebSphere Application Server extended J2EE security role based access control to protect the WebSphere Application Server administrative subsystem. Four administrative roles have been defined to provide degrees of authority needed to perform certain WebSphere Application Server administrative functions from either the Web-based administrative console or the system management scripting interface. The authorization policy is only enforced when global security is enabled. The four administrative security roles are defined as follows:

- *Monitor*: The least privileged role which basically allows a user to view the WebSphere Application Server configuration and current state.
- *Configurator*: This role has monitor privilege plus the ability to change the WebSphere Application Server configuration.
- *Operator*: This role has monitor privilege plus the ability to change runtime state, such as starting or stopping services for example.
- *Administrator*: This role has operator and configuration privilege and the permission that is required to access sensitive data including server password, LTPA password and keys, and so on.

Task 1: Enabling Global Security

You will enable global security within WebSphere. Then you will configure access to the Administrative console by defining a number of roles and mapping those roles to existing users. Then, in order to test your configuration, you will attempt to perform various functions from the various users and see if the security configurations correctly limits your ability to perform those functions.

__ 1. Verify that your WAS instance has been started.

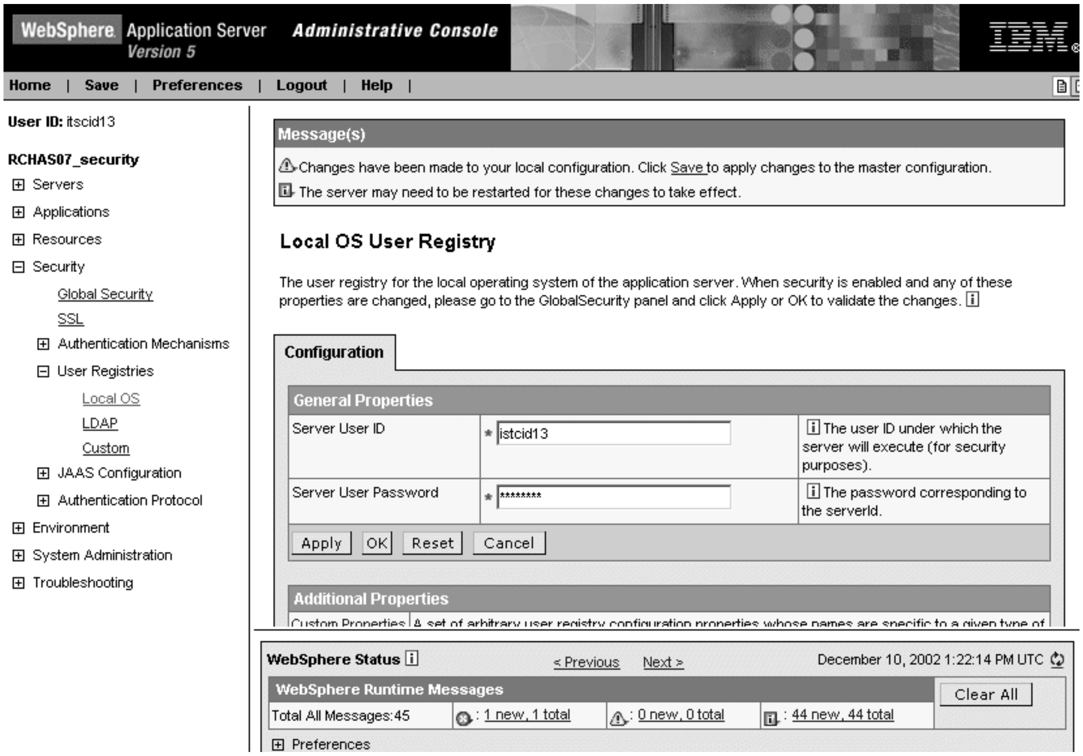
__ 2. Start the Admin Console and view local configuration information:

i. In a browser, specify the URL `http://<hostName>:port/admin`

ii. In the Login window, specify your team's ID for **User ID** and click **OK**

__ 3. Click the plus sign (+) to the left of **Security** to expand the tree. Now expand the **User Registries** and click **Local OS**.

__ 4. In the right pane we need to give a userID and password (needs to be for a privileged user), so that WebSphere can authenticate against the local registry. Enter your team's ID as **Server User Id** and `java1a` for **Server User Password** (see Figure 5-1).



User ID: itscid13

Message(s)

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.

Local OS User Registry

The user registry for the local operating system of the application server. When security is enabled and any of these properties are changed, please go to the GlobalSecurity panel and click Apply or OK to validate the changes.

General Properties		
Server User ID	* istcid13	The user ID under which the server will execute (for security purposes).
Server User Password	* ********	The password corresponding to the serverId.

Additional Properties

Custom Properties | A set of arbitrary user registry configuration properties whose names are specific to a given type of

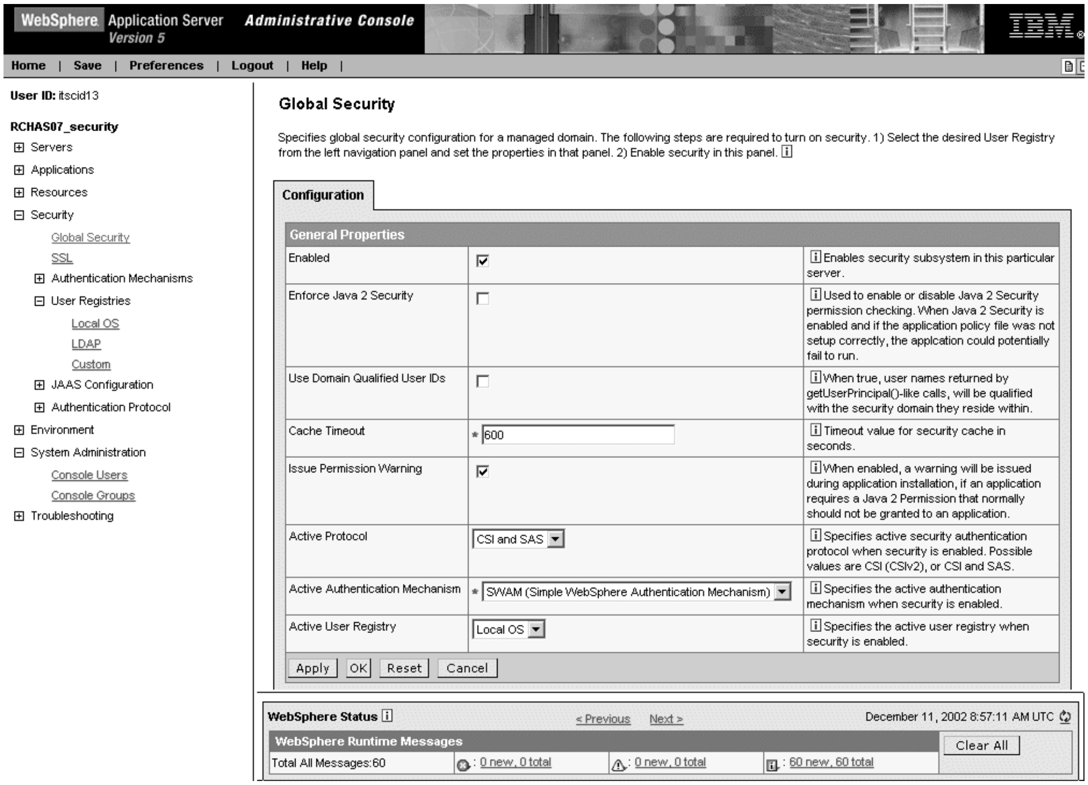
WebSphere Status < Previous Next > December 10, 2002 1:22:14 PM UTC

WebSphere Runtime Messages			
Total All Messages:45	: 1 new, 1 total	: 0 new, 0 total	: 44 new, 44 total

Figure 5-1 Specifying secure user ID

__ 5. Click **OK**.

__ 6. A new frame appears. In the right pane, look for the checkbox labeled **Enabled** and click it. This will check the **Enforce Java 2 Security** box automatically. Ensure that **Active User Registry** is set to **LocalOS** (see Figure 5-2).



Global Security

Specifies global security configuration for a managed domain. The following steps are required to turn on security. 1) Select the desired User Registry from the left navigation panel and set the properties in that panel. 2) Enable security in this panel.

Configuration

General Properties		
Enabled	☑	Enables security subsystem in this particular server.
Enforce Java 2 Security	☐	Used to enable or disable Java 2 Security permission checking. When Java 2 Security is enabled and if the application policy file was not setup correctly, the applcation could potentially fail to run.
Use Domain Qualified User IDs	☐	When true, user names returned by getUserPrincipal()-like calls, will be qualified with the security domain they reside within.
Cache Timeout	* 600	Timeout value for security cache in seconds.
Issue Permission Warning	☑	When enabled, a warning will be issued during application installation, if an application requires a Java 2 Permission that normally should not be granted to an application.
Active Protocol	CSI and SAS	Specifies active security authentication protocol when security is enabled. Possible values are CSI (CSIv2), or CSI and SAS.
Active Authentication Mechanism	* SWAM (Simple WebSphere Authentication Mechanism)	Specifies the active authentication mechanism when security is enabled.
Active User Registry	Local OS	Specifies the active user registry when security is enabled.

WebSphere Status December 11, 2002 8:57:11 AM UTC

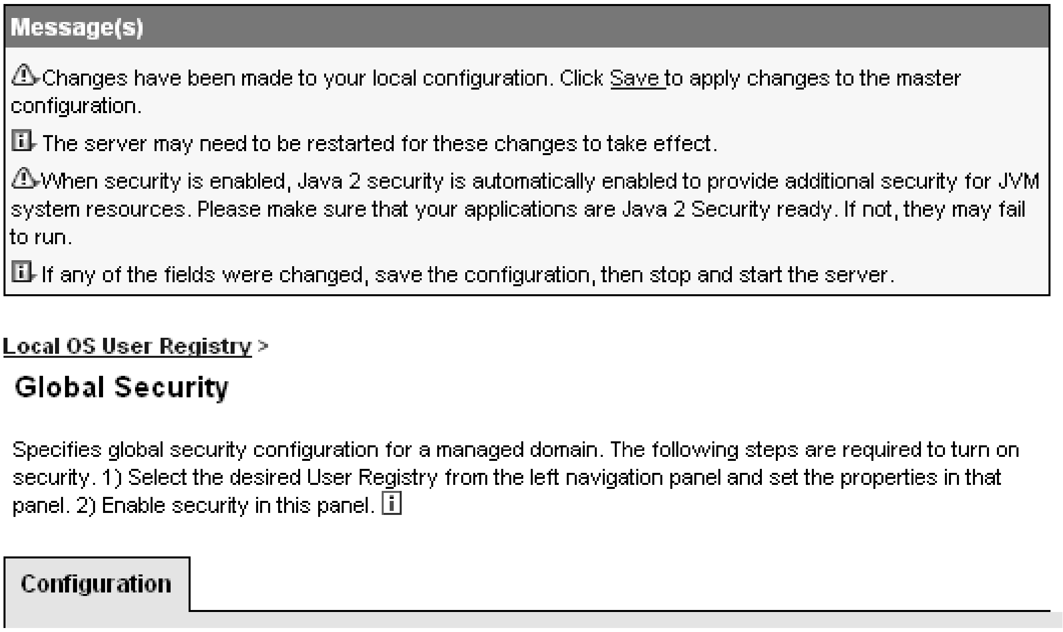
WebSphere Runtime Messages

Total All Messages:60 | ⊗ : 0 new, 0 total | ⚠ : 0 new, 0 total | ⓘ : 60 new, 60 total

Figure 5-2 Enabling global security

__ 7. Click **OK**.

Note: Check the message panel at the top of the page. Any errors should be reviewed and corrected before continuing (see Figure 5-3).



Message(s)

⚠ 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.

⚠ When security is enabled, Java 2 security is automatically enabled to provide additional security for JVM system resources. Please make sure that your applications are Java 2 Security ready. If not, they may fail to run.

ⓘ If any of the fields were changed, save the configuration, then stop and start the server.

Local OS User Registry >

Global Security

Specifies global security configuration for a managed domain. The following steps are required to turn on security. 1) Select the desired User Registry from the left navigation panel and set the properties in that panel. 2) Enable security in this panel.

Configuration

Figure 5-3 Checking the messages

__ 8. When no errors are shown in the message panel, click **Save** in the menu bar, then click the **Save** button to save the master configuration.

__ 9. From the Admin Console go to **System Administration** and select **Console Users**.

__ 10. Click **Add**.

__ 11. Now map each of the roles to the following userIDs (click OK for each user ID; see Figure 5-4):

- your team's ID -> Administrator
- config -> Configurator
- operator -> Operator
- monitor -> Monitor

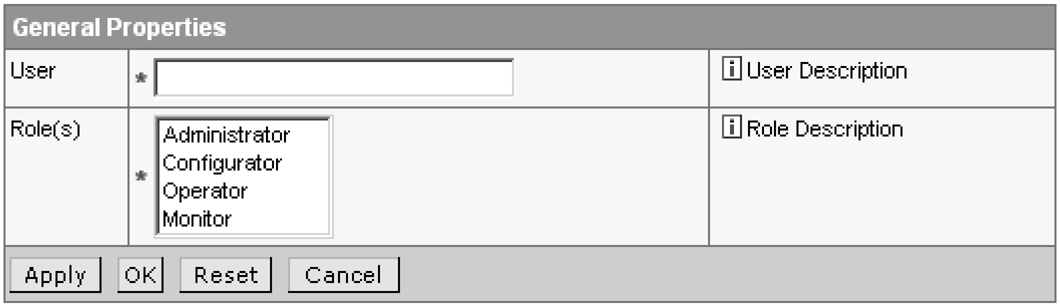
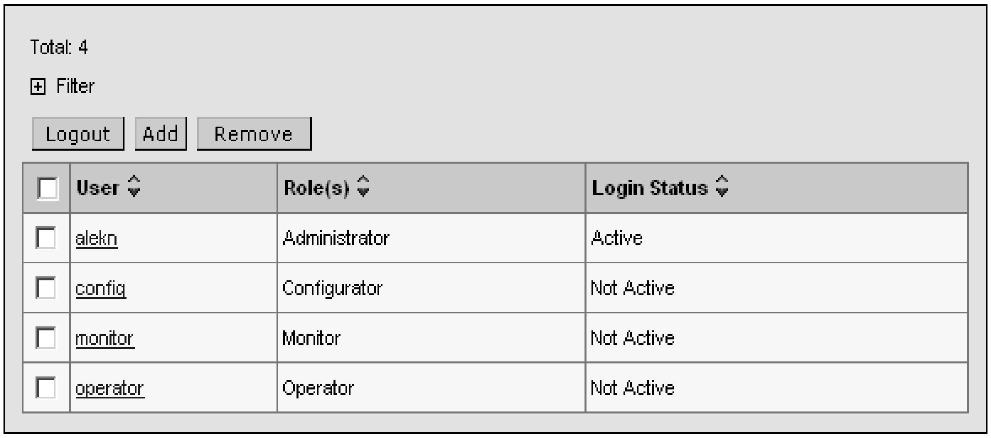


Figure 5-4 Mapping predefined roles to the user IDs

__ 12. After finishing this mapping you should see a screen similar to Figure 5-5.



□	User	Role(s)	Login Status
□	alekn	Administrator	Active
□	config	Configurator	Not Active
□	monitor	Monitor	Not Active
□	operator	Operator	Not Active

Figure 5-5 Created console users

__ 13. Save your changes.

__ 14. Now we need to click **Logout** in the menu bar.

__ 15. In order for the Global Security to take effect, stop and start your team's WebSphere Application Server instance:

i. From the Qshell command line execute the following 2 commands:

```
cd /qibm/proddata/webas5/base/bin
```

```
stopServer -instance Teamx TeamxSvr
```

Note: Server name is case sensitive.

ii. Wait for the completion of the last command and start your instance again:

```
startServer -instance Teamx TeamxSvr
```

Important: Next time you need to use the stopServer script, after restarting an application server with enabled security, you will have to specify a user ID and password (this parameters have been set in step __ 4. on 43):

```
stopServer -instance Teamx TeamxSvr -username Teamx -password java1a
```

__ 16. Wait for the confirmation message that the server has been started.

__ 17. Point your browser to the following URL:

```
http://<hostName>:port/admin
```

__ 18. You should see a login prompt with 2 boxes: User ID and Password (see Figure 5-6).

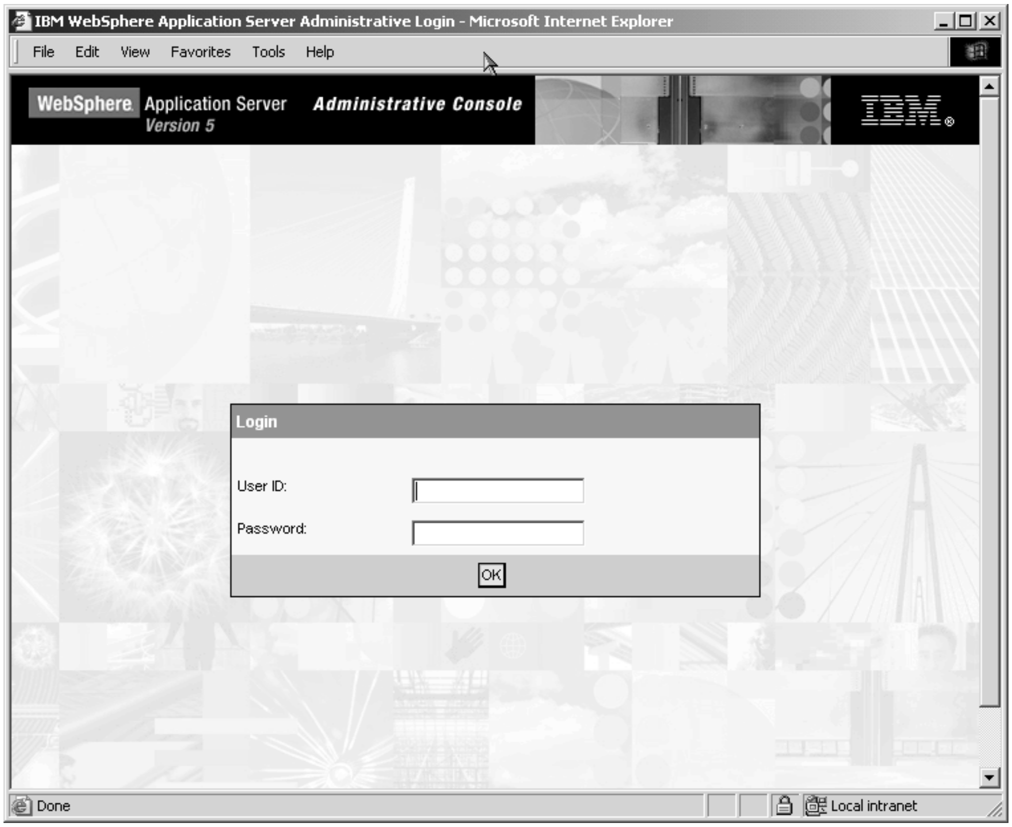

IBM WebSphere Application Server Administrative Login - Microsoft Internet Explorer
File Edit View Favorites Tools Help
WebSphere Application Server Version 5
Administrative Console
IBM
Login
User ID:
Password:
OK
Done
Local intranet

Figure 5-6 Login prompt

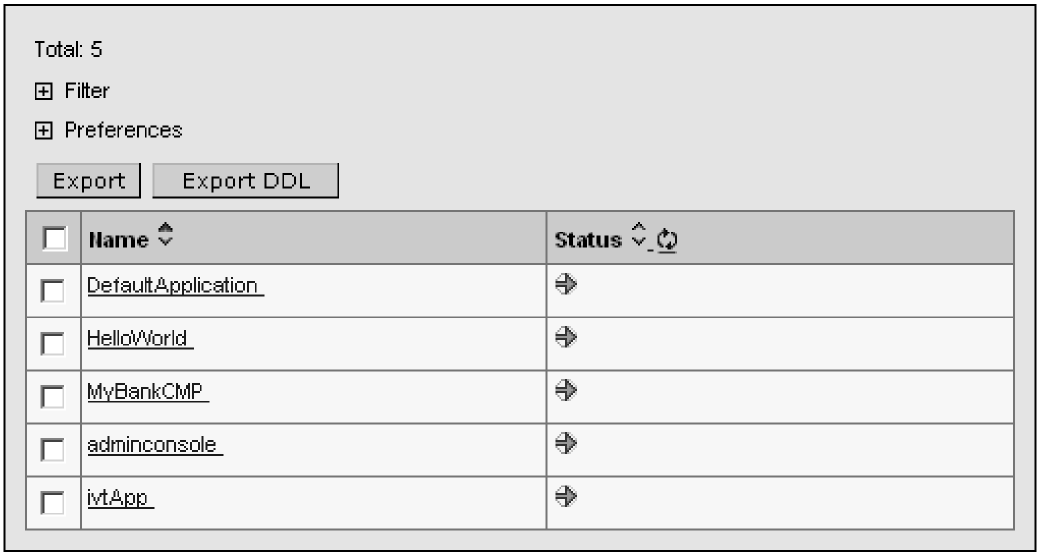
__ 19. Now we need to test access through 4 users:

- Your team’s ID

- config
- operator
- monitor

__ 20. Notice that each user's console will have different functions available. Go ahead and experiment with each one, starting, stopping, and exporting various applications.

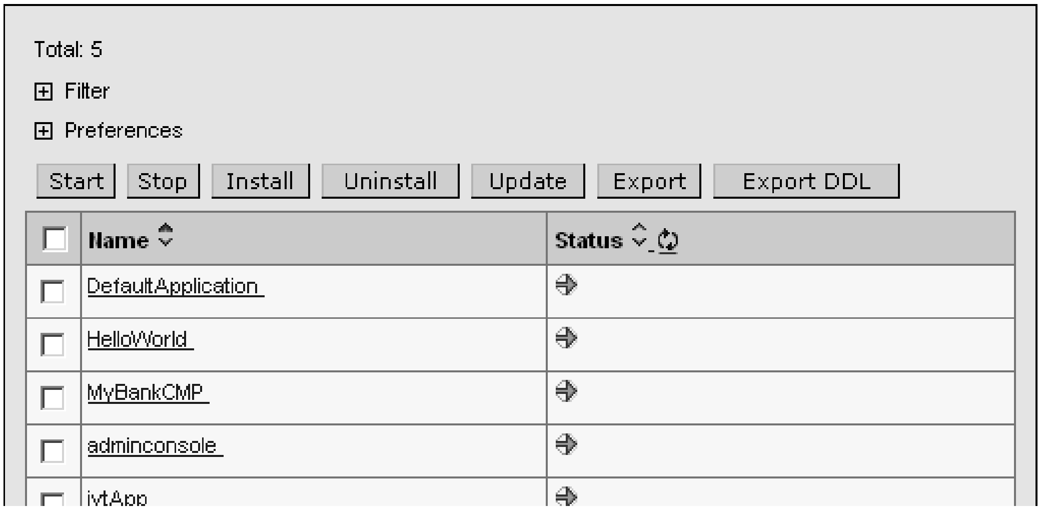
i. Start a new Admin Console and authenticate with the **monitor** user ID (monitor/java1a). Notice that when you look at **Enterprise Applications** under **Applications**, you have very few abilities. You can not even start or stop an application. The only two functions listed are **Export** and **Export DDL** (see Figure 5-7).



□	Name	Status
□	DefaultApplication	➜
□	HelloWorld	➜
□	MyBankCMP	➜
□	adminconsole	➜
□	ivtApp	➜

Figure 5-7 Options available for user ID "monitor"

ii. Start another Admin Console session and authenticate with the admin user (<yourTeamID>/java1a). Notice that when you look at **Enterprise Applications** under **Applications** that you have a whole lot more functionality available (see Figure 5-8).



□	Name	Status
□	DefaultApplication	➔
□	HelloWorld	➔
□	MyBankCMP	➔
□	adminconsole	➔
□	ivtApp	➔

Figure 5-8 Options available for user ID "Administrator"

iii. Start another Admin Console session and authenticate with the Configurator user (config/java1a). Notice that when you look at **Enterprise Applications** under **Applications** that you have a different set of functionality available.

iv. Start another Admin Console session and authenticate with the Operator user (operator/java1a). Notice that when you look at **Enterprise Applications** under **Applications** that you have yet another set of functionality available.