Monitoring IBM WebSphere Application Server

eG Innovations Product Documentation

www.eginnovations.com

Total Performance Visibility

Table of Contents

Table of Figures

Chapter 1: Introduction

IBM WebSphere Application Server, a software application server, is the flagship product within IBM's WebSphere brand. WAS is built using open standards such as J2EE, XML, and Web Services. It works with a number of web servers including Apache HTTP Server, Netscape Enterprise Server, Microsoft Internet Information Services (IIS), IBM HTTP Server for i5/OS, IBM HTTP Server for z/OS, and IBM HTTP Server for AIX/Linux/Microsoft Windows/Solaris. It delivers the secure, scalable, resilient application infrastructure you need for a Service Oriented Architecture (SOA).

Like any other application server, performance setbacks suffered by the WebSphere application server too can affect the availability of critical services it supports. To avoid such an eventuality, you need to continuously monitor the performance of the WebSphere application server.

The eG Enterprise suite facilitates 24x7 monitoring of the WebSphere application server and proactive alerting of probable error conditions detected on the server. Since different versions of the WebSphere application server vary in both capabilities and architecture, eG Enterprise employs different mechanisms to monitor the various versions. This is why, eG Enterprise provides two separate monitoring models for WebSphere Application server – one for version 4.0 (or 5.1) of the server, and another for version 6.0 and above. While it uses the component-type WebSphere Application – 4/5.x to monitor the former, the component-type WebSphere Application is used for monitoring the latter. This chapter discusses both these models in great detail.

This document engages you in an elaborate discussion on how eG Enterprise monitors each of the popular web application servers in the market.

Chapter 2: How does eG Enterprise Monitor IBM WebSphere Application Server 4/5.x?

The eG Enterprise suite provides two different monitoring models for WebSphere Appplication server 4.0 (or 5.1) and the WebSphere Application server 6.0 and above. While it uses the component-type WebSphere Application – 4/5.x to monitor the former, the component-type WebSphere Application is used for monitoring the latter. The steps for configuring and monitoring both these models are handled in this section.

This section deals with the procedures and practices of configuring and monitoring a WebSphere Application Server 4.0 (or 5.1).

2.1 Configuring WebSphere Application Server 4.0 to work with the eG Agent

To monitor a WebSphere Application server 4.0, a specific WebSphere monitoring eG component has to be installed on it. In order to configure a WebSphere Application server for eG monitoring, do the following:

1. Start the WebSphere Application server, by typing the command **WAS_HOME/bin/adminserver.sh**, where **WAS_HOME** is the home directory of the WebSphere Application server.
2. Start the WebShere Advanced Administrative Console using the command **adminclient.sh** (see Figure 2.1).

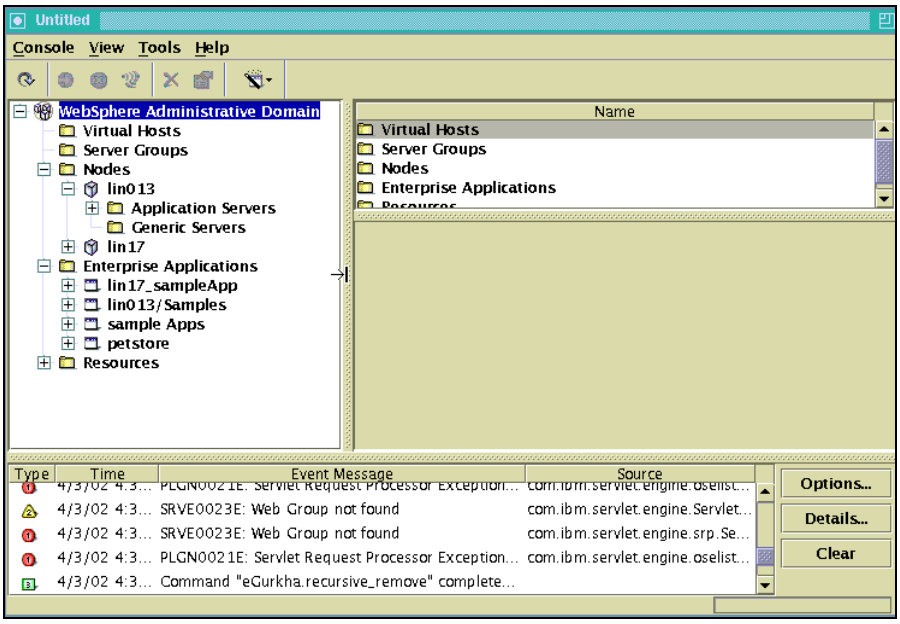

Figure 2.1: The WebSphere Advanced Administrative Console

3. In order to install the eG component, right-click on **Enterprise Applications** in the left pane and select **Install Enterprise Application** from the menu that pops up as shown in Figure 2.2 below:

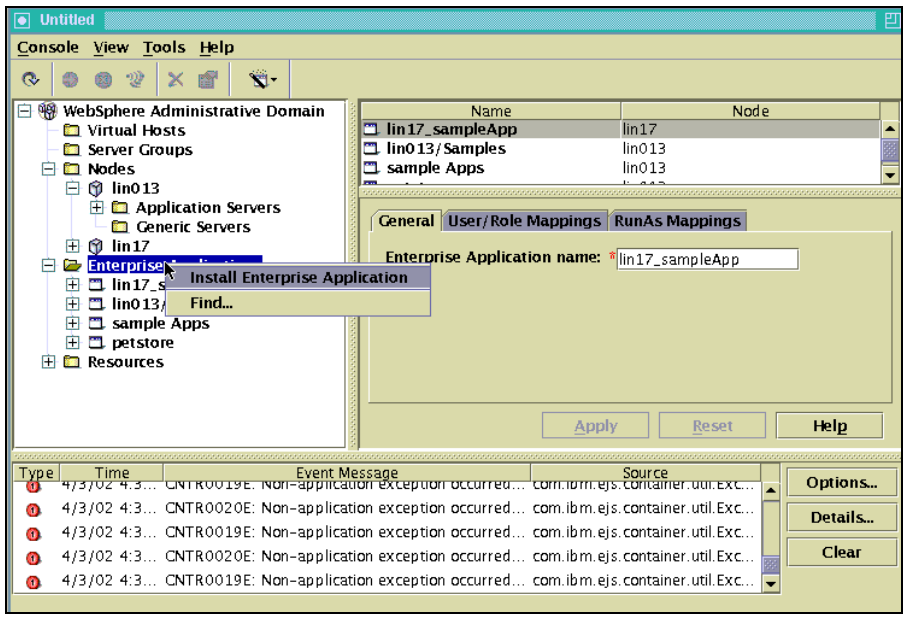

Figure 2.2: Selecting the option for installing an enterprise application

4. In the dialog box that appears (see Figure 2.3), select the **Install Application (*.ear)** option and

specify the **Path** to the **egurkha.ear** file (which will be located in <EG_HOME_DIR>/egurkha/lib directory, where <EG_HOME_DIR> is the directory in which eG has been installed). The Browse button can also be used for locating the file. Next, name the new component as **egurkha** by specifying the same against the **Application Name** text box. Then, click the **Next >** button in the dialog box.

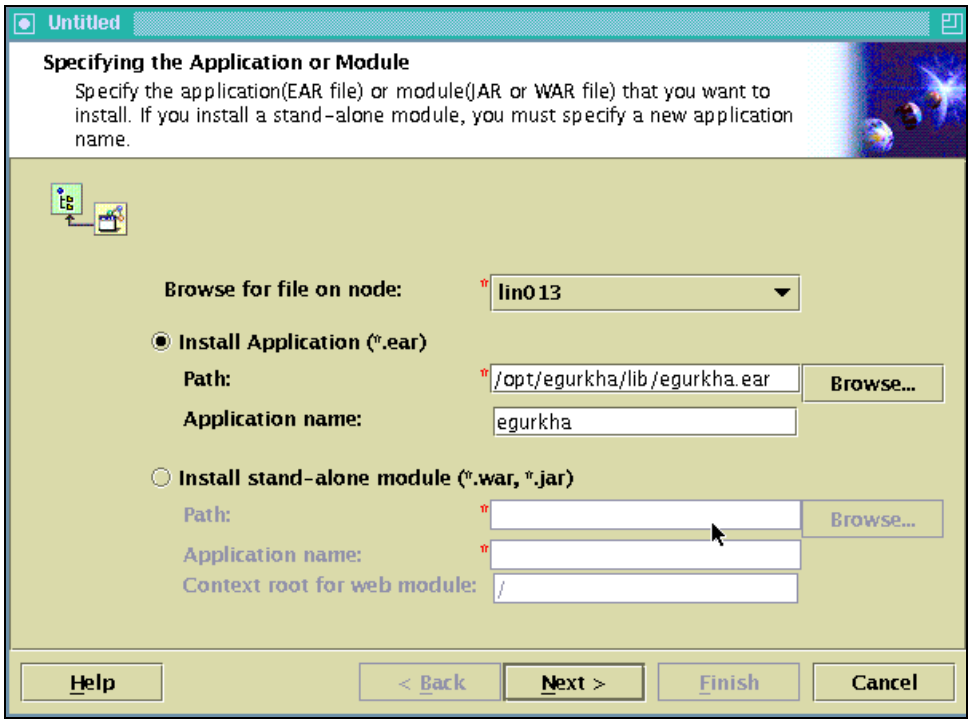

Figure 2.3: Specifying the application or module to be installed

5. As Mapping of users to roles is not required for our component, click the **Next >** button in the **Mapping Users to Roles** dialog box (see Figure 2.4).

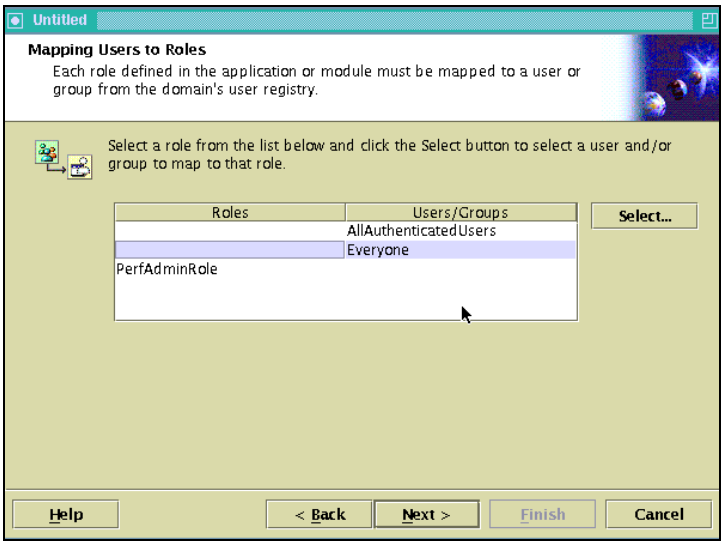

Figure 2.4: Mapping users to roles

6. The dialog box that appears on clicking the **Next >** button is the **Mapping EJB RunAs Roles to Users** dialog box (see Figure 2.5). As this mapping also does not apply to the eG component, click the **Next >** button in the dialog box to proceed with the installation.

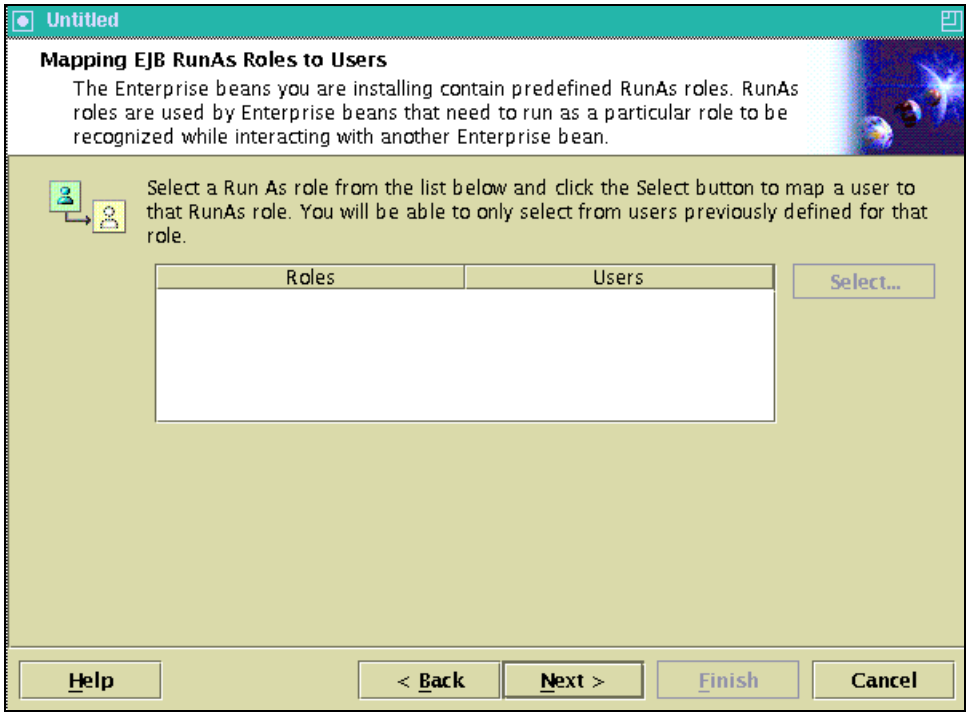

Figure 2.5: Mapping EJB RunAs roles to users

7. Once you arrive at **the Binding Enterprise Beans to JNDI Names** dialog box (see Figure

2.6), click the **Next >** button and move on.

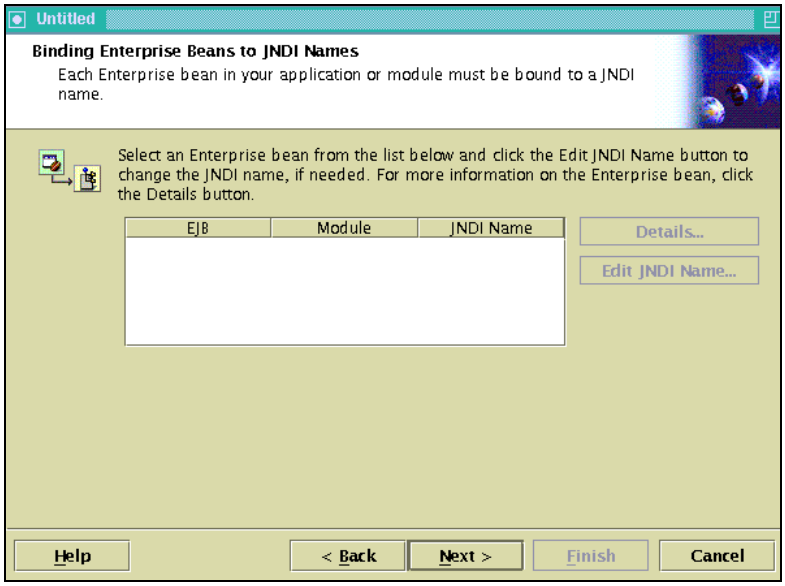

Figure 2.6: Mapping Enterprise Beans to JNDI Names

8. This will take you to the **Mapping EJB References to Enterprise** *Beans* dialog box shown by Figure 2.7 below, where you would have to click the **Next >** button and navigate to the next step of the installation process.

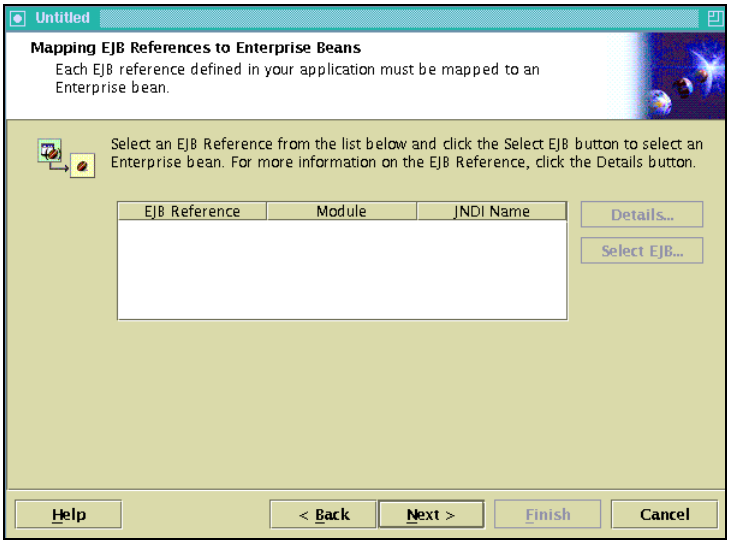

Figure 2.7: Mapping EJB references to enterprise Beans

9. The **Mapping Resource References to Resources** dialog box depicted by Figure 2.8 below will then come up. Click the **Next >** button in this dialog box to proceed.

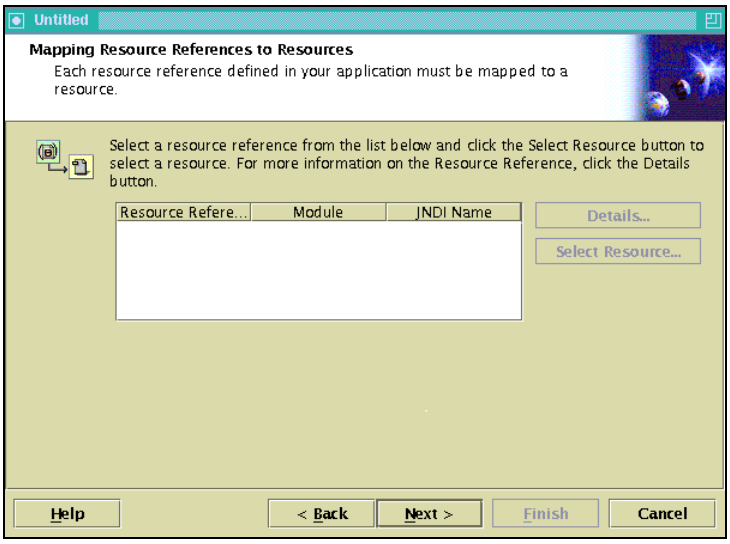

Figure 2.8: Mapping resource references to resources

10. In the **Specifying the Default Datasource for EJB Modules** dialog box (see Figure 2.9) that appears, click the **Next >** button to move to the next step.

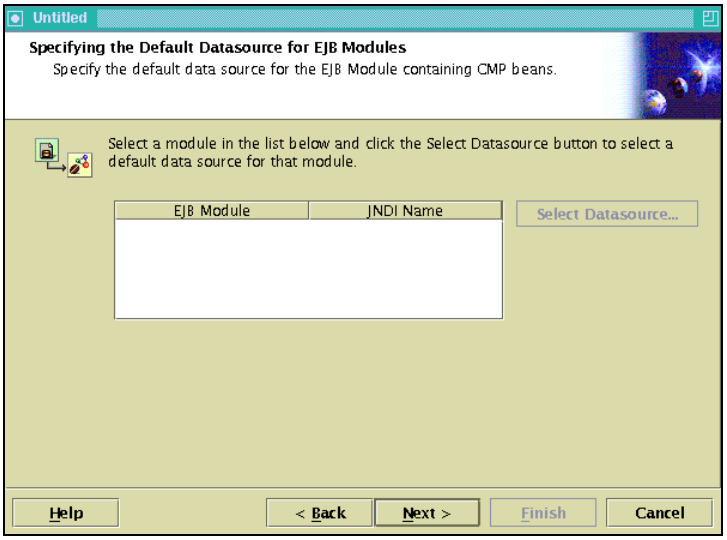

Figure 2.9: Specifying the default datasource for EJB modules

11. The next dialog box depicted by 2.1 below will prompt you to specify the Datasource for individual CMP Beans. Once again, click the **Next >** button to proceed.

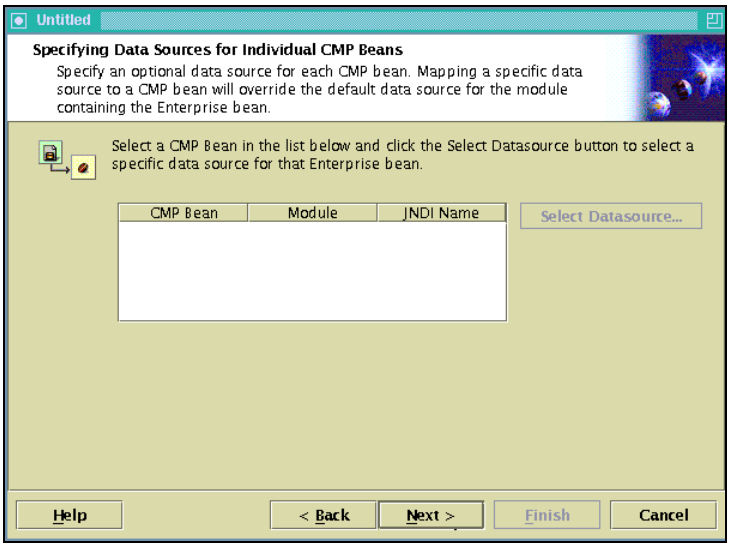

Figure 2.10: Specifying data sources for individual CMP Beans

12. In the dialog box that appears soon after, you need to select the virtual hosts on which the modules within the new component are to be installed. Just click the **Next >** button in the dialog box and move on.

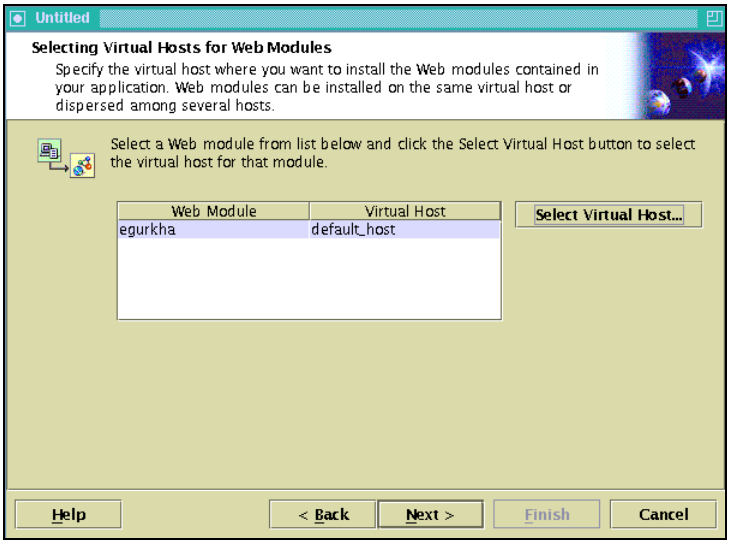

Figure 2.11: Selecting virtual hosts for Web modules

13. The next dialog box (Figure 2.12) is where we select the application server on which the component **egurkha** is to be installed. For that, first, click the **Select Server…** button.

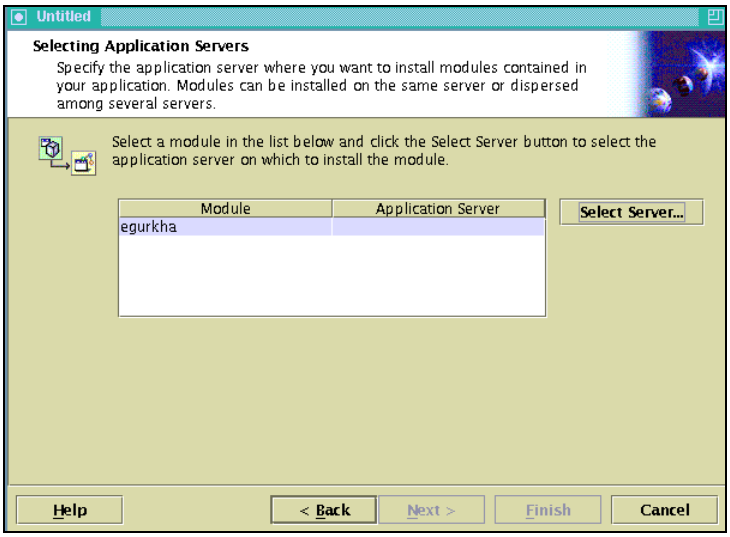

Figure 2.12: Selecting component servers

14. When this is done, the following dialog box will appear listing the Component Servers available and the nodes to which they are connected. Select the server on which the new component is to be installed and click the **OK** button in the dialog box.

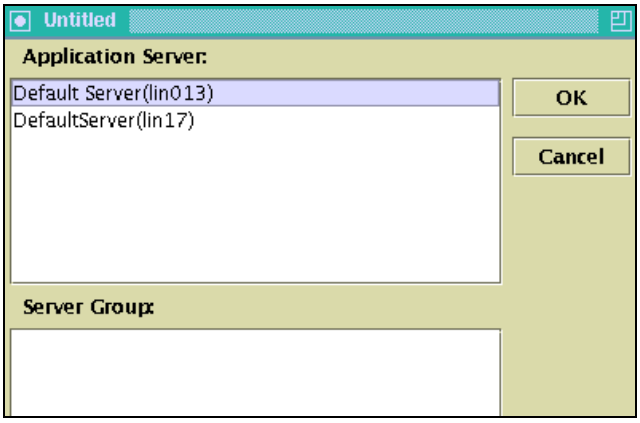

Figure 2.13: Component servers

15. You will then return to the **Selecting Component Servers** dialog box. Here, you can see the application server you selected appear against the **Module egurkha** as shown in Figure 2.14. Click the **Next >** button to view a summary of the settings that you had defined.

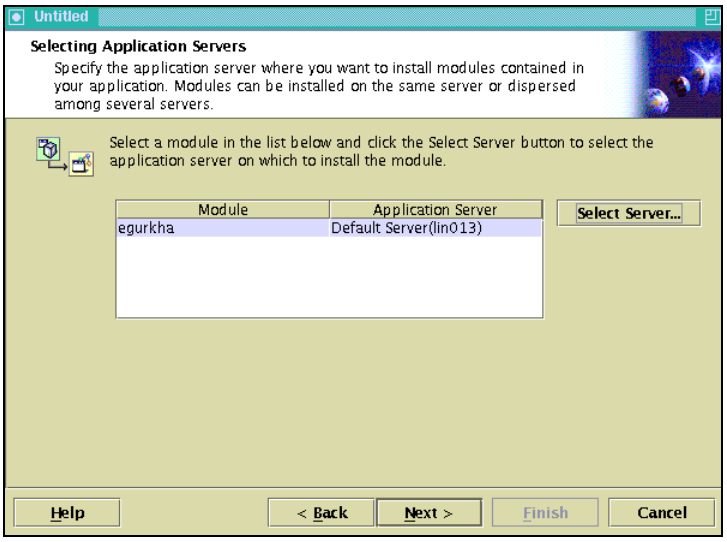

Figure 2.14: Viewing the selected application server

16. After viewing the summary information, end the installation by clicking the **Finish** button in the **Completing the Application Installation Wizard** dialog box.

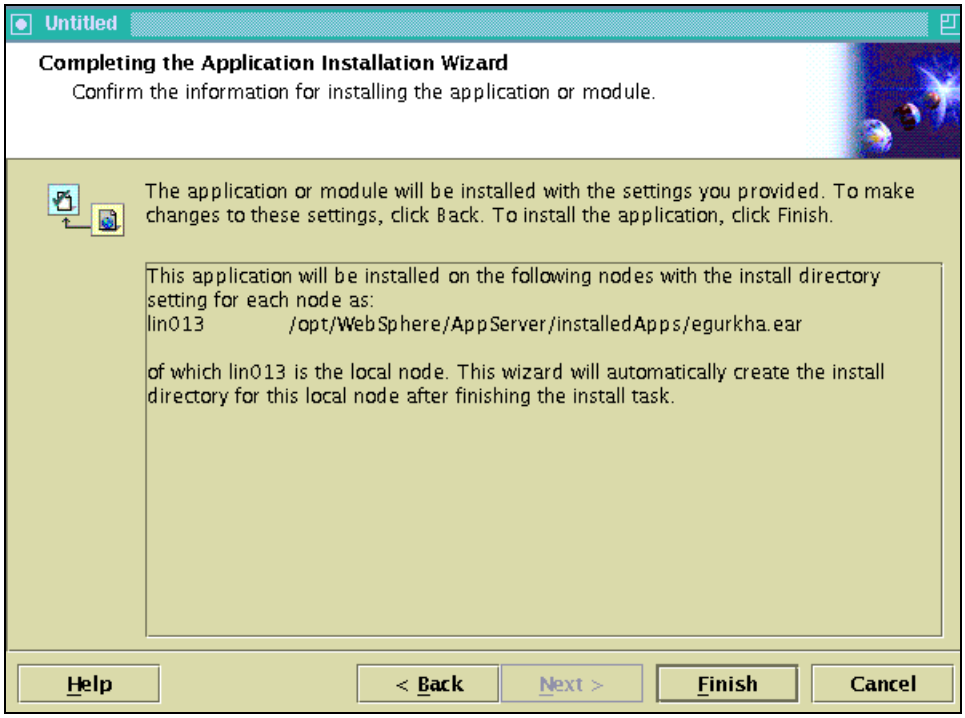

Figure 2.15: Completing the component installation

17. Click the **OK** button in the message box that will pop up on successful installation of the eG component.

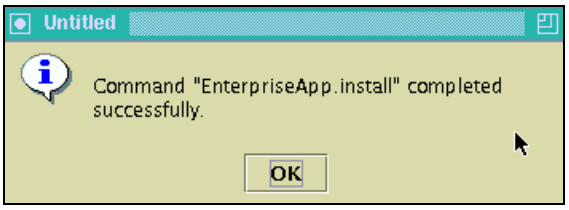

Figure 2.16: Message box informing the user of successful installation

18. Clicking **OK** will take you back to the Administrative Console. Refresh the contents of the console by clicking the **Refresh** button on the tool bar.

19. Then, regenerate the Webserver plugin by expanding the **Nodes** option in the tree-structure, selecting the node on which you installed the new component, right-clicking on it and picking the **Regen Webserver Plugin** option from the shortcut menu (see Figure 2.17). Doing so will update the Webserver Plugin with the changes made to the configuration of the WebSphere Application Server.

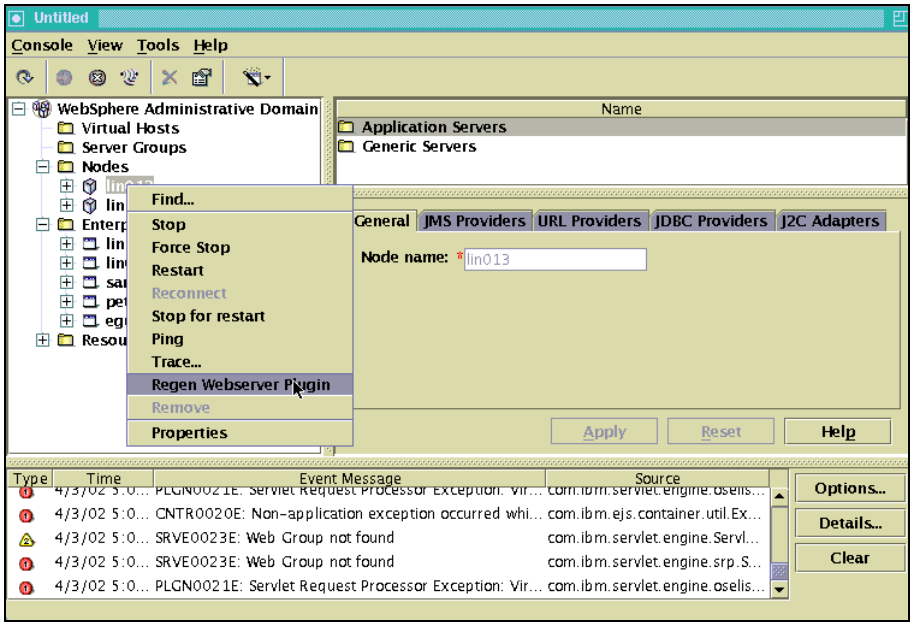

Figure 2.17: Selecting the Regen Webserver Plugin option

20. Then, to ensure that the changes made take effect, stop the application server (if it has been started already) and then restart it. In order to do this, expand **Nodes** in the tree-structure, select the application server on which the new component **egurkha** has been installed, right-click on it, and select **Stop** from the pop-up menu. This sequence has been depicted in Figure 2.18 below. Later, right-click on the server once again, but this time, select **Start** from the pop-up menu.

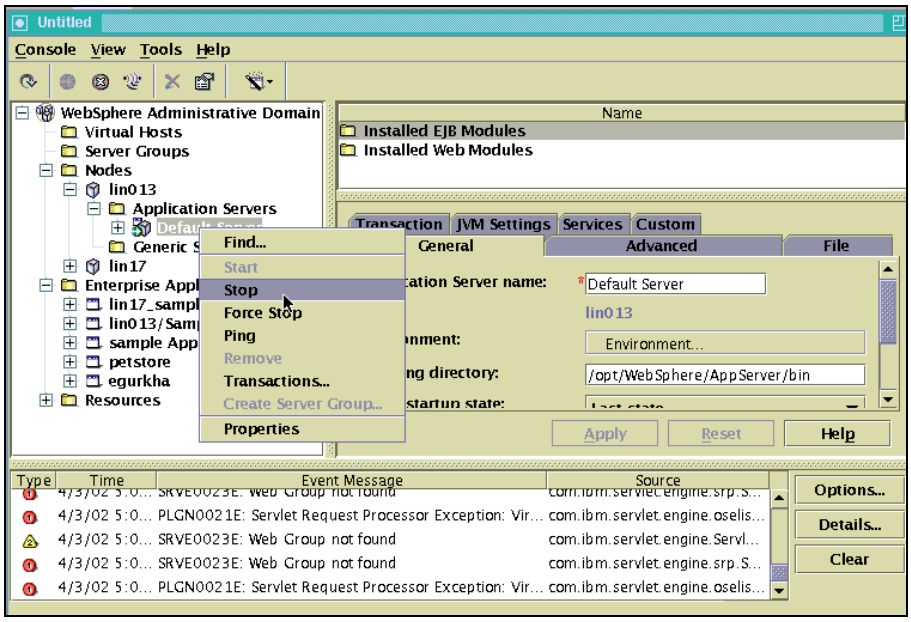

Figure 2.18: Restarting the application server

2.2 Running the Application

Now, to run the application, follow the steps given below:

1. Open the Administrative Console by following steps 1 and 2 of Section **Chapter 2**.
2. Then, expand **Enterprise Applications** in the tree-structure, select the **egurkha** application,

right-click on it and choose **Start** from the menu (see Figure 2.19).

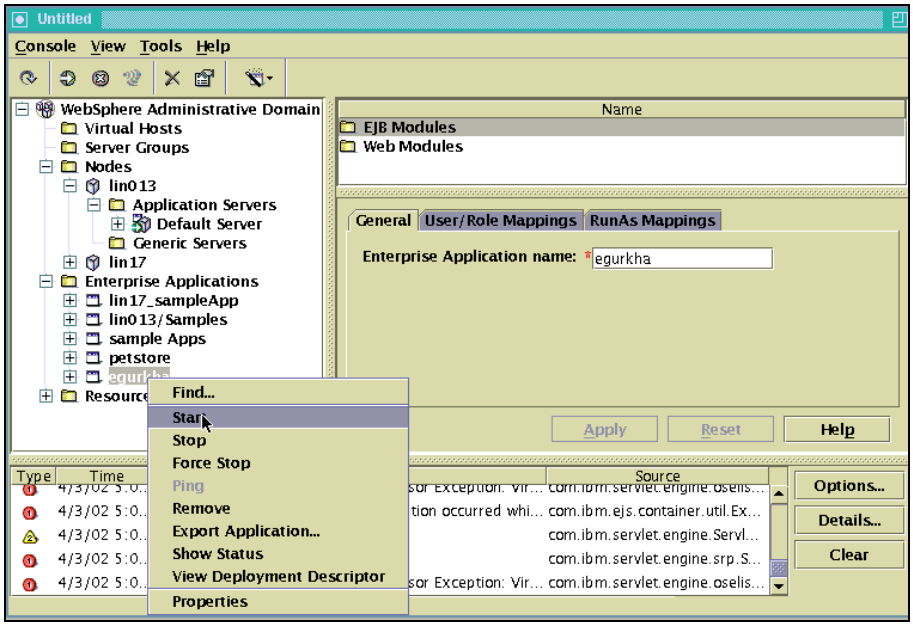

Figure 2.19: Running the installed application

2.3 Setting Instrumentation Levels

The resources in a WebSphere administrative domain are instrumented so that statistical data can be collected. Instrumentation refers to the mechanism by which some aspect of the running system is measured. Each resource category has an instrumentation level that determines which measurements are available to be collected for that category.

Before you commence monitoring of WebSphere Application servers, the instrumentation levels of resource categories need to be set.

To do so, follow the steps given below:

1. In the Administrative Console, highlight your application server.
2. Then, from the series of tabs listed in the right pane, select **Services** as shown in Figure 2.20.

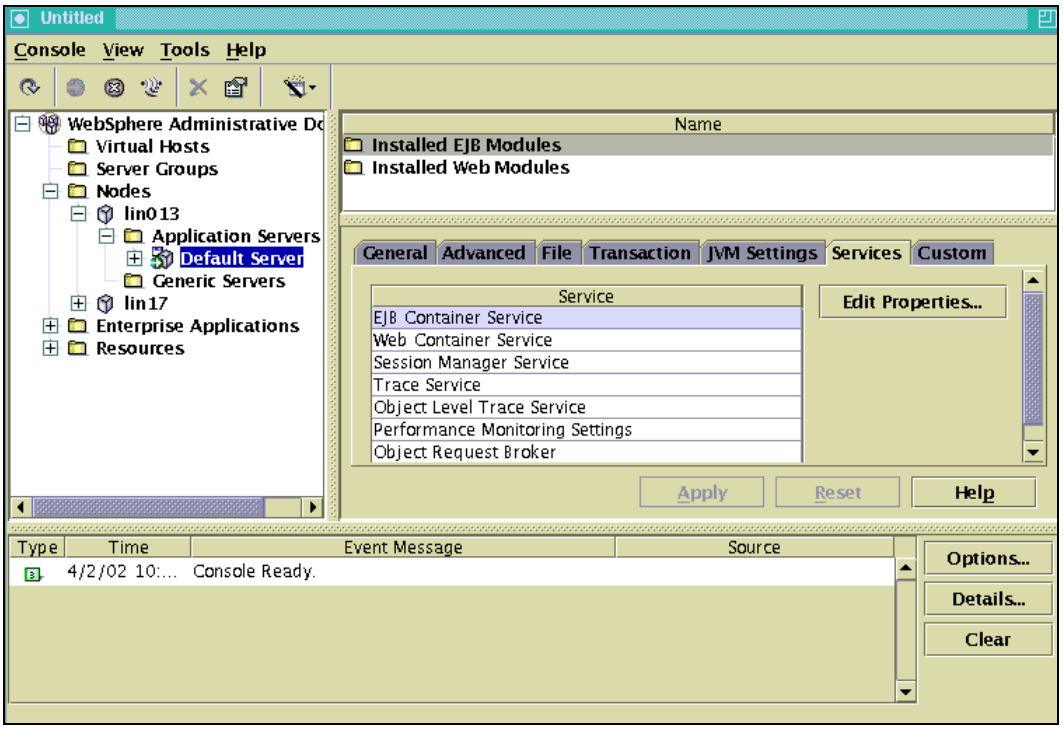

Figure 2.20: Selecting the Services tab for the application server

3. From the Services displayed therein, choose **Performance Monitoring Settings** and click the **Edit Properties**… button as depicted by Figure 2.21.

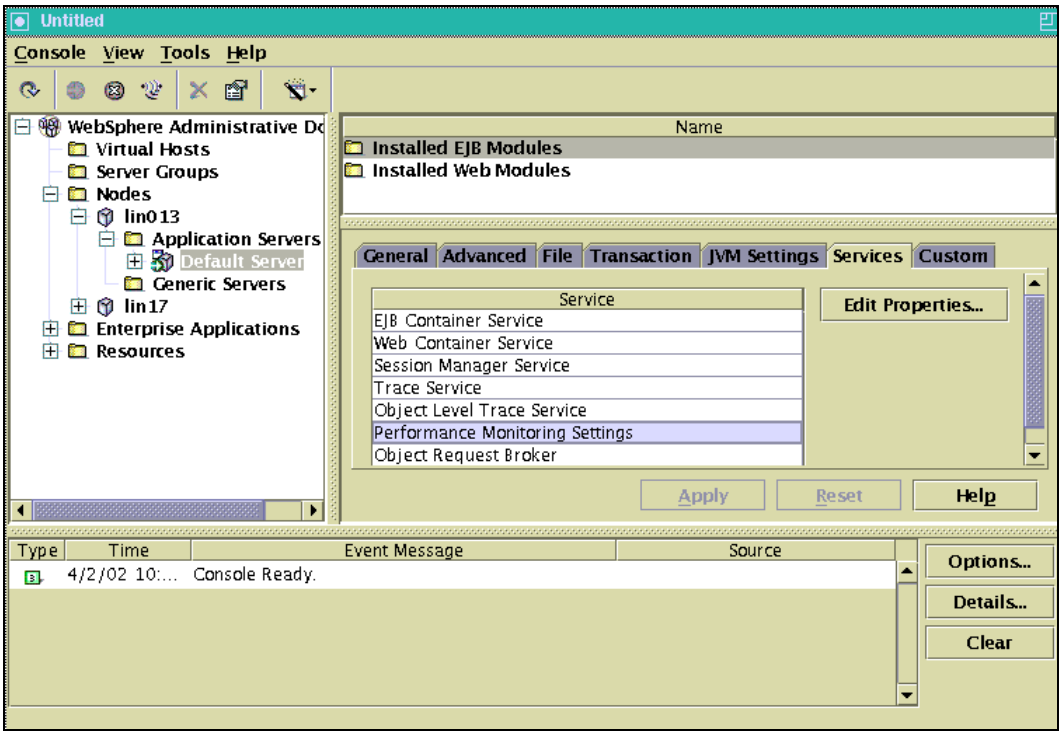

Figure 2.21: Selecting the Performance Monitoring Settings service

4. Doing so, will reveal the following dialog box:

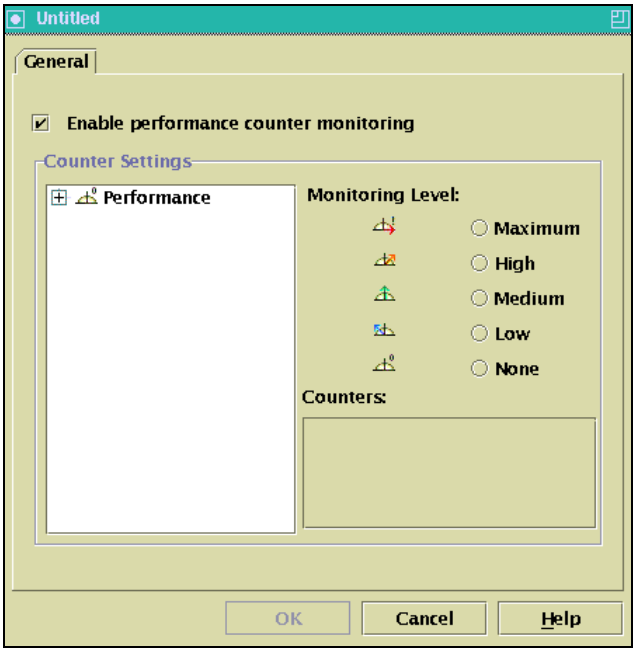

Figure 2.22: Editing the performance monitoring settings

5. Expanding the **Performance** node here will bring to light the resource categories whose performance can be instrumented (see Figure 2.23). To instrument a resource category, select it and then choose **High** as the **Monitoring Level** of the same. This ensures that all the key measurements are collected.

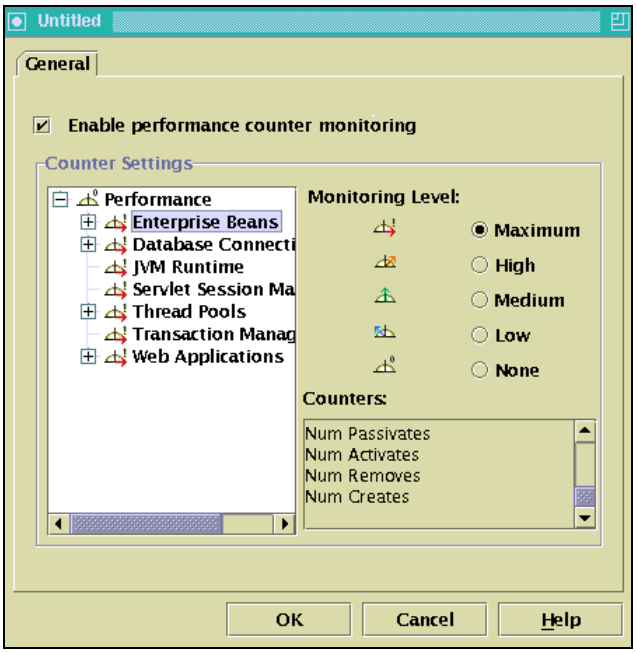

Figure 2.23: Viewing resource categories

6. Choosing "**High**" as the monitoring level is a must for eG to Monitor your WebSphere Application Server.

2.4 Configuring a WebSphere Application Server 5.1 to work with the eG Agent

To configure the WebSphere application server 5.1, do the following:

1. Open Internet Explorer, and type the following URL: http://<WebSphereIP>:<WebSpherePort>/, to connect to the WebSphere Application server's administration console.
2. A **Login** dialog box requesting a **User Id** will appear (see Figure 2.24).

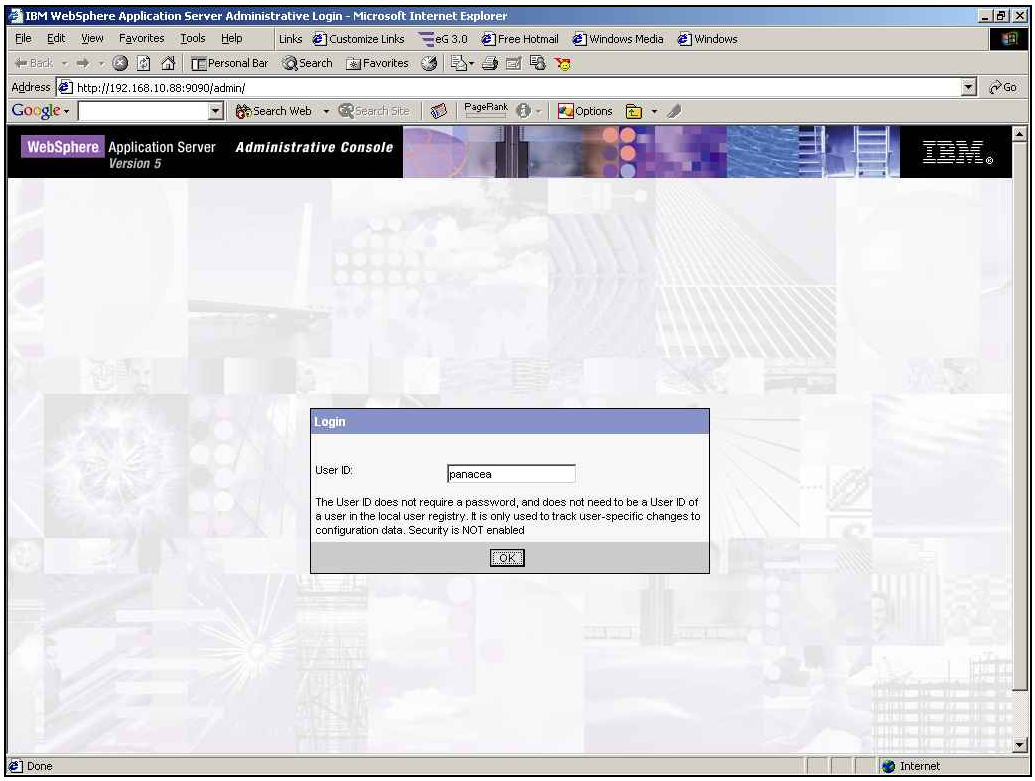

Figure 2.24: Specify the login user id

3. In Figure 2.24, specify a user id to login to the admin console, and click the **OK** button.
4. In the screen that appears next (see Figure 2.25), select the **Install New Application** link in the left pane, to install a new application on the WebSphere server.

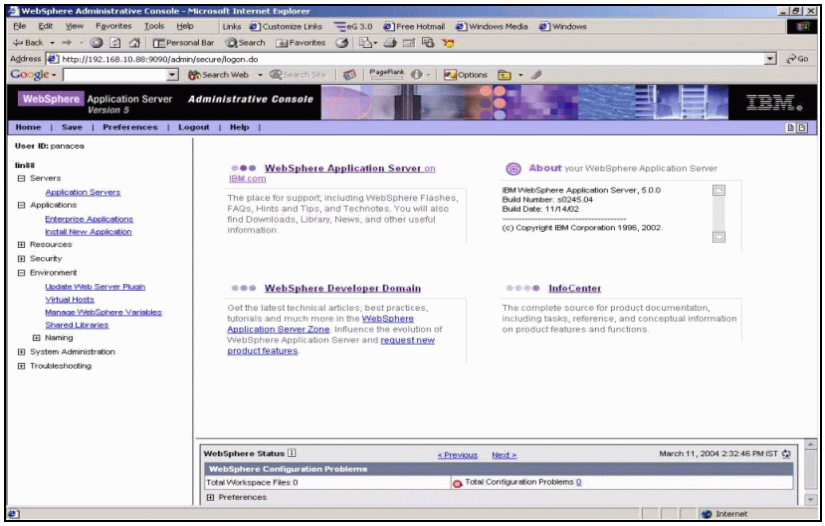

Figure 2.25: Installing a new application on the WebSphere server

5. Upon clicking the link, Figure 2.26 will appear. Select the **Local path** option in the right pane of Figure 2.26, and specify the full path to the location of the **egurkha.ear** file that is to be deployed on the WebSphere server.

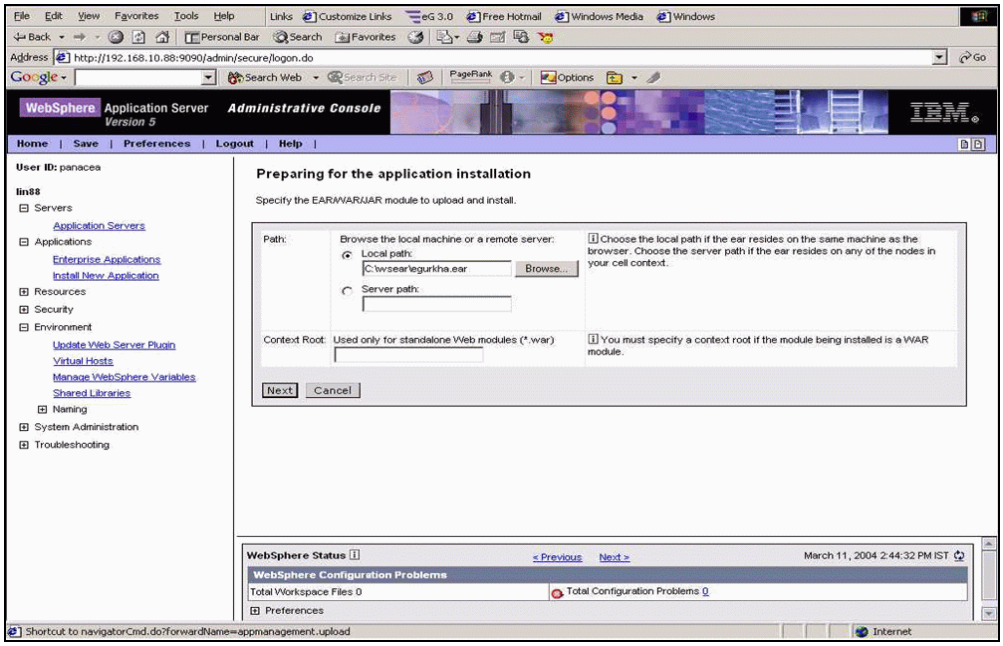

Figure 2.26: Specifying the path to the egurkha.ear file

6. Then, click the **Next** button in Figure 2.26 to proceed with the deployment.

7. Once the **Next** button (in Figure 2.26) is clicked, Figure 2.27 will appear. Select the **Generate Default Bindings** check box in the right pane of Figure 2.27 and click the **Next** button to proceed.

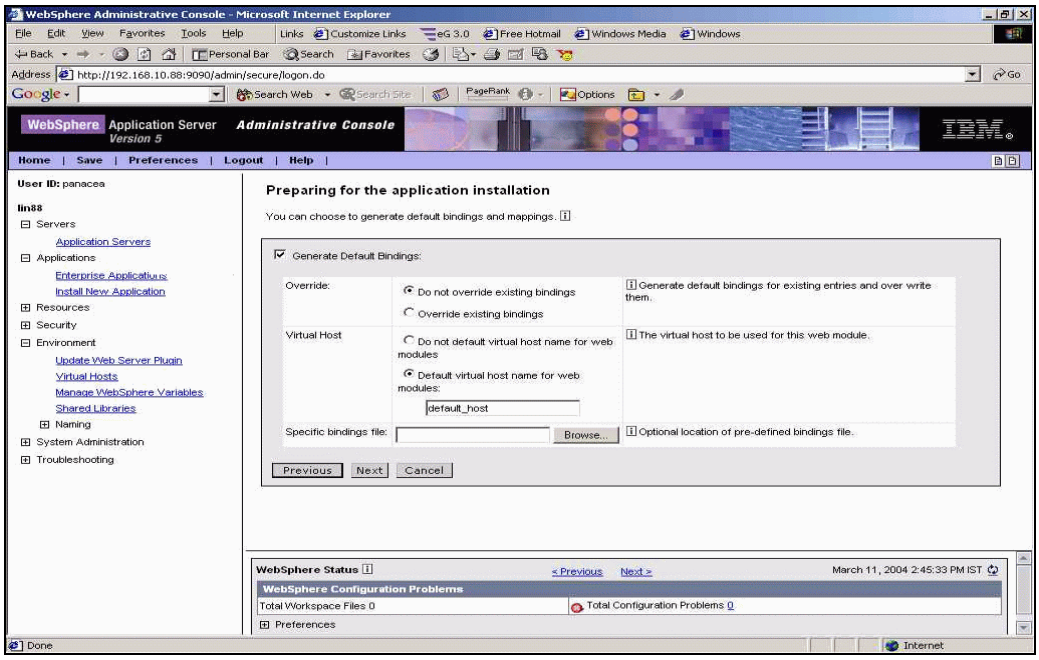

Figure 2.27: Select the Generate Default Bindings option

8. In the right pane of Figure 2.28 that appears, provide **egurkha** as the **Application Name** and click the **Next** button.

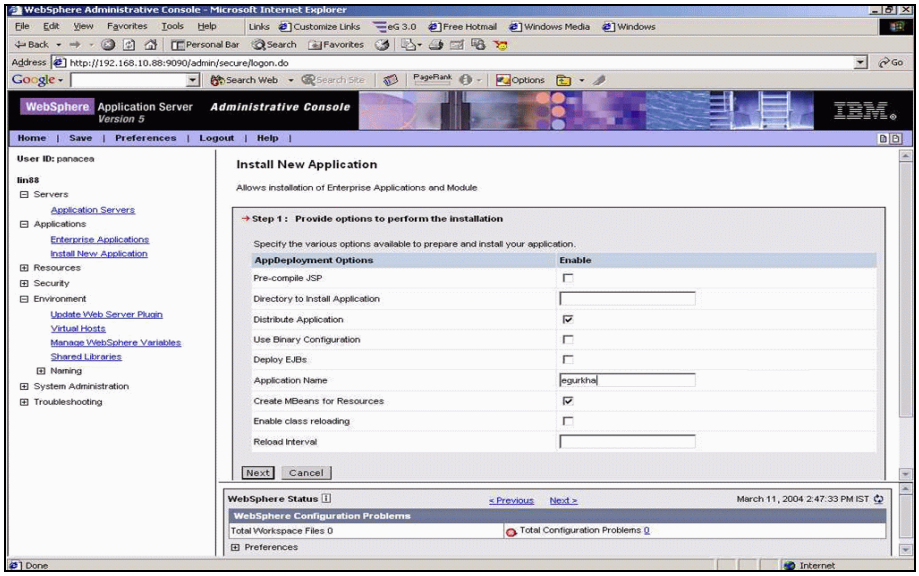

Figure 2.28: Specifying the name of the application.

9. Next, map the modules in the application with the servers. Since our application consists of only one module, **egurkha**, the same will be displayed under the **Module** column in the right pane of

Figure 2.29. To map the **egurkha** module with the current server, select the check box alongside it, and click the **Next** button.

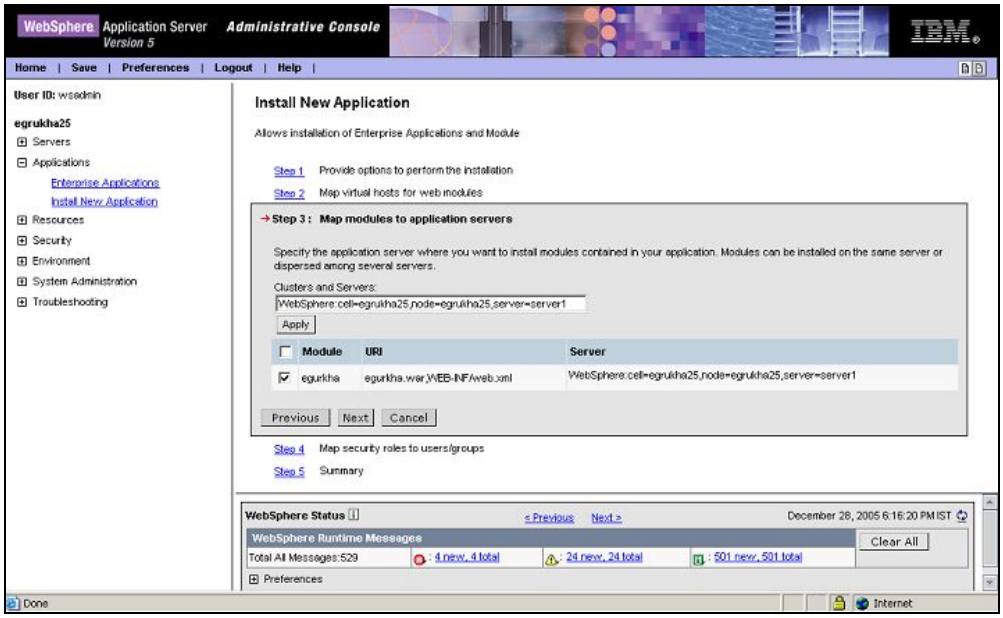

Figure 2.29: Mapping the egurkha module with the current server

10. Next, you will be prompted to map security toles to users/groups (see Figure 2.30). If security has been enabled for the WebSphere server, then this mapping is mandatory. If not, simply click on the **Next** button in Figure 2.29 to switch to the next step of the installation. In the case where security is enabled, it is recommended that you always map the *administrator* role with a specific *admin user/group* for monitoring purposes. To achieve this, first select the **administrator** role displayed in Figure 2.30 by clicking on the check box alongside it.

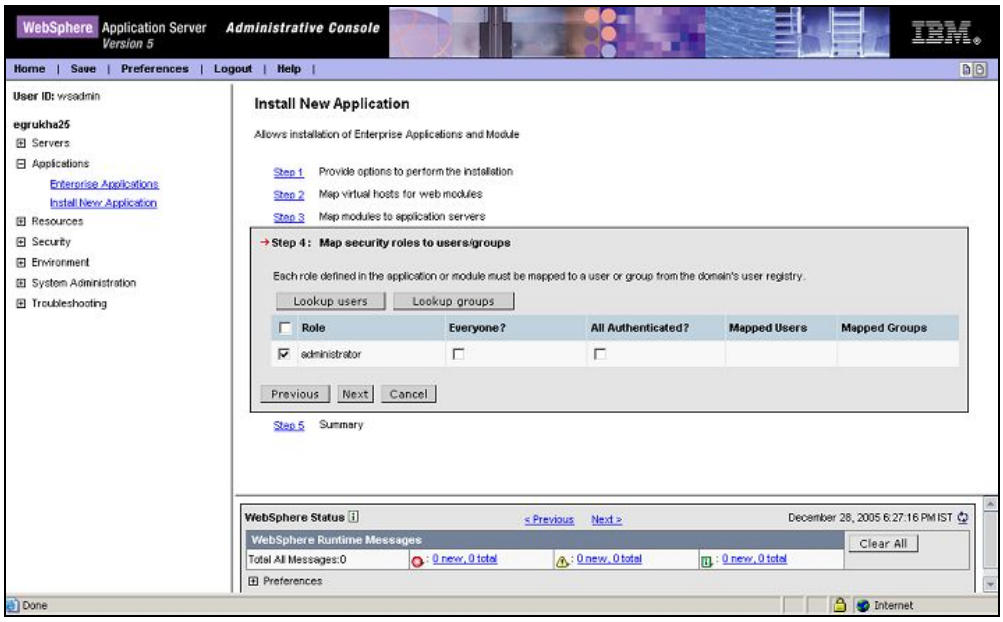

Figure 2.30: Selecting the security role

Note:

To determine whether security has been enabled for a WebSphere server or not, try logging into the administrative console of the WebSphere server. If you are requested for a user name and password, it is a clear indication that security has been enabled. On the contrary, if you are prompted only for a User ID (as in the case of Figure 2.24), then it denotes that security has not been enabled.

11. Then, click on the **Lookup users** button in Figure 2.30 to associate the role with an administrator user or group. Figure 2.31 will then appear.

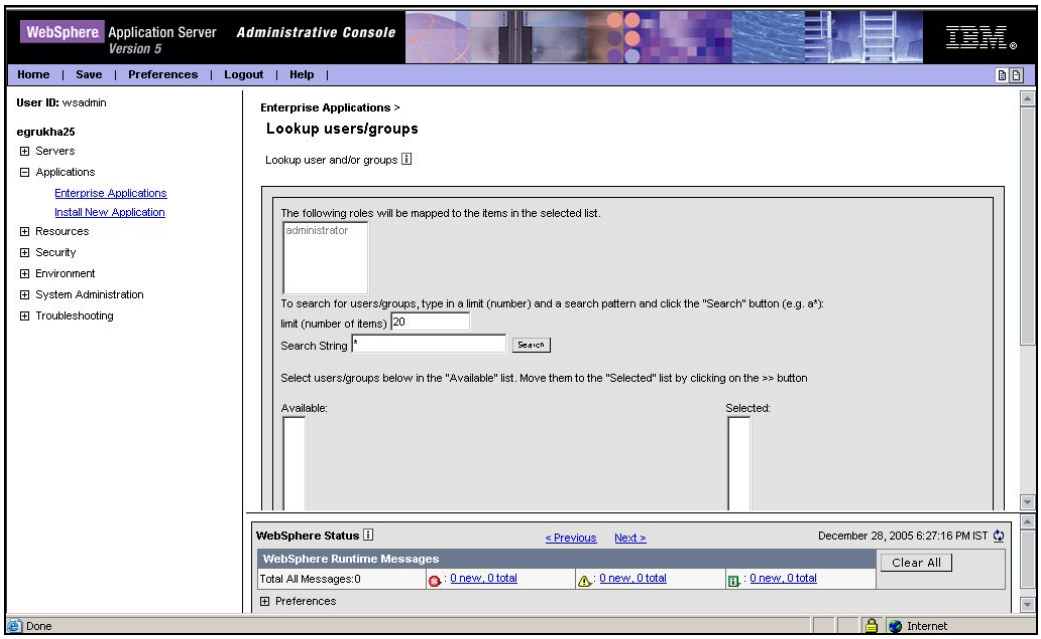

Figure 2.31: Associating the security role with a user/group

12. In the **Search String** text box of Figure 2.31, specify the user ID that you want to search for and map to the *administrator* role. Then, click on the **Search** button (see Figure 2.32).

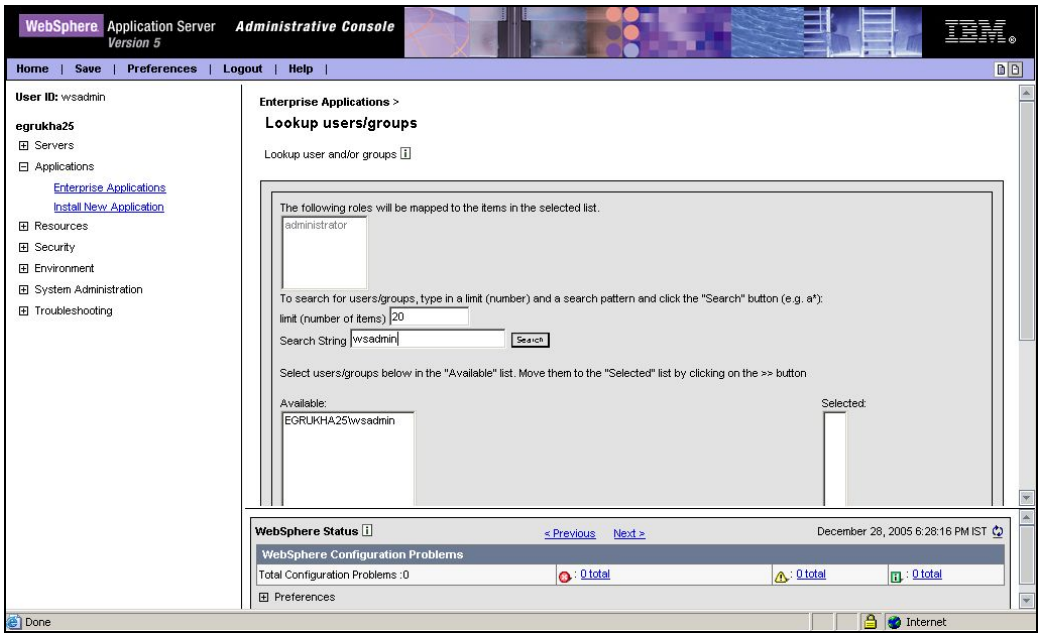

Figure 2.32: Searching for the user

13. All user IDs that match the specified **Search String** will appear in the **Available** list box (see

Figure 2.32). From this list box, select the users to be mapped to the *administrator* role, and click on the **>>** button to transfer the selection to the **Selected** list (see Figure 2.33 and Figure 2.34). Then, click on the **OK** button in Figure 2.33.

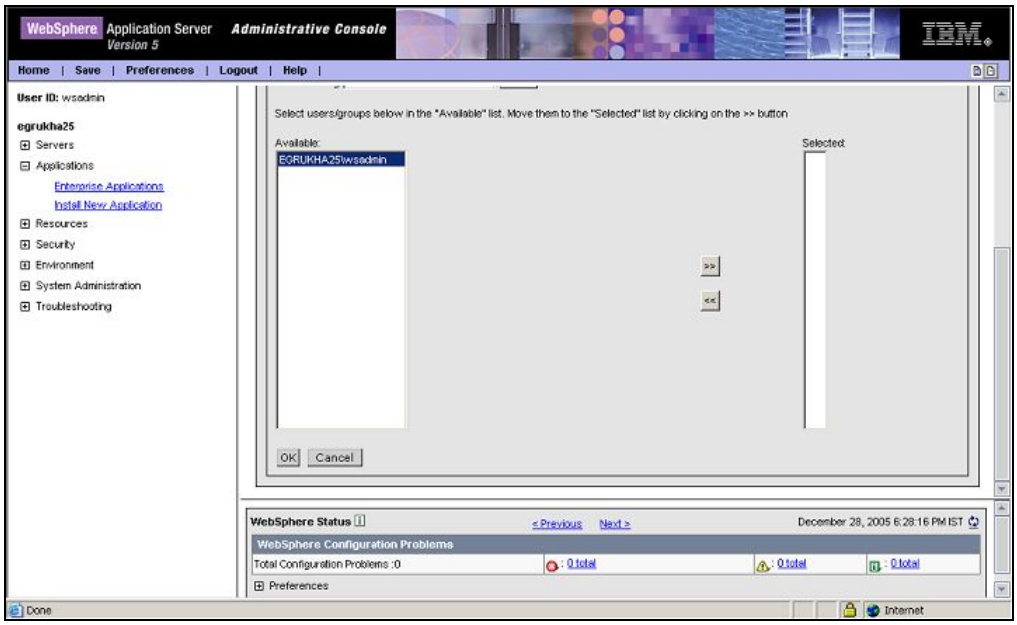

Figure 2.33: Selecting the users to be mapped to the administrator group

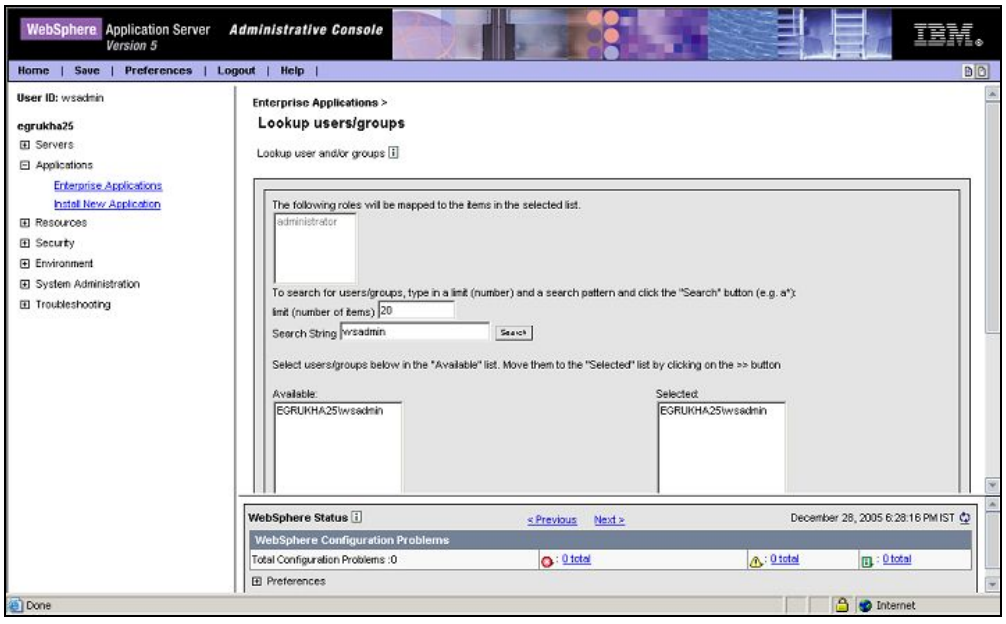

Figure 2.34: Figure 1.34: Applying the association

14. You will then return to Figure 2.30, where the selected users will appear against the

administrator role (see Figure 2.35). Click on the **Next** button to proceed.

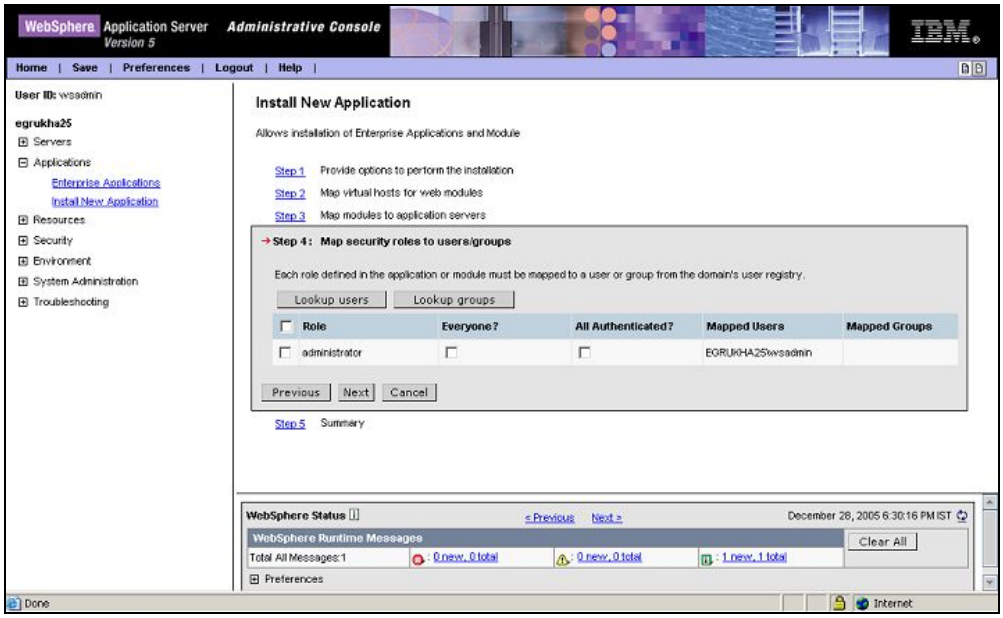

Figure 2.35: Selected users mapped to the administrator role

15. View a summary of the chosen install options (see Figure 2.36), and click the **Finish** button to begin installation.

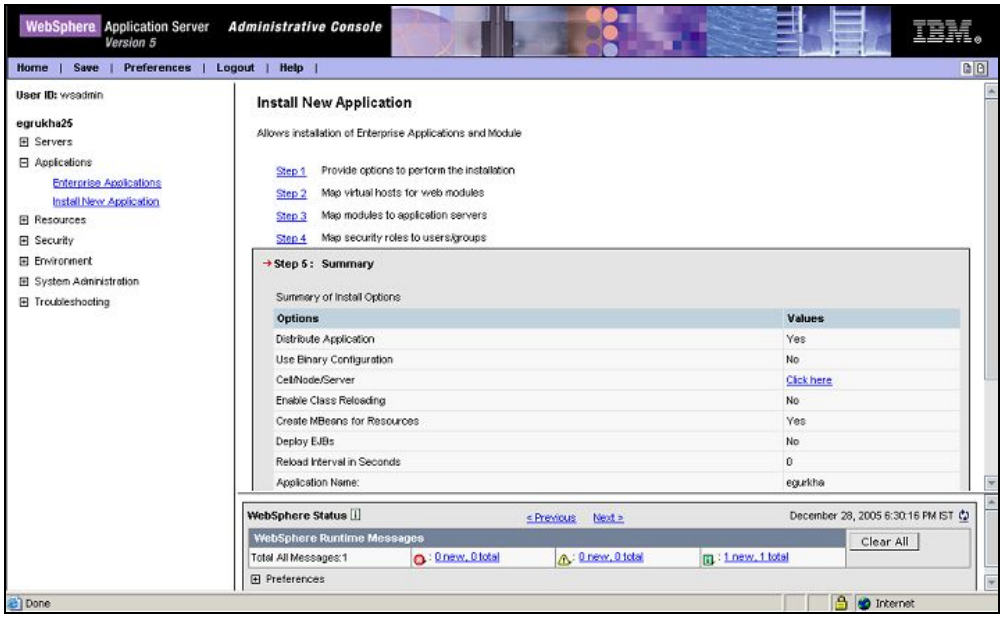

Figure 2.36: A summary of the install options

16. The subsequent screen (Figure 2.37) will indicate the successful installation of the **egurkha**

application. Now, click on the **Save to Master Configuration** link in the right pane of this screen.

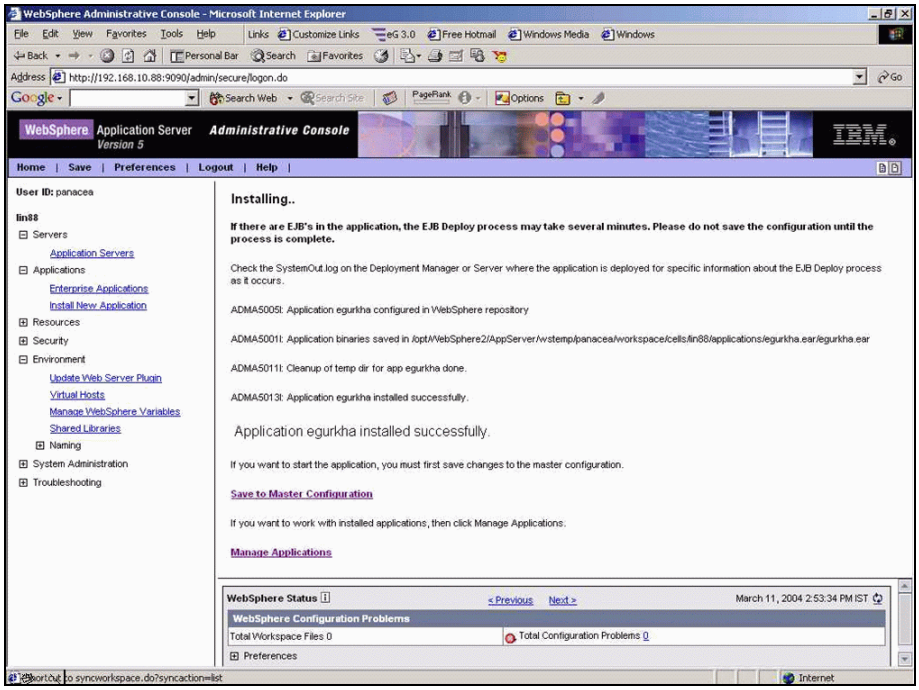

Figure 2.37: electing the Save to Master Configuration link

17. Next, click on the **Save** button in Figure 2.38 that appears to update the master repository with the changes.

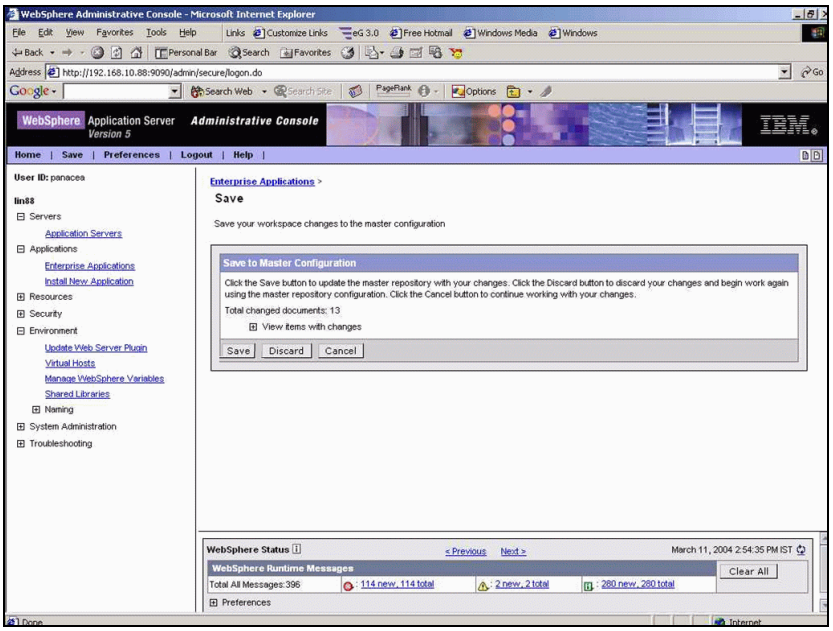

Figure 2.38: Updating the master repository

18. After updating the master configuration, click on the **Update Web Server Plugin** link under the **Environment** section in the left pane of Figure 2.38. Then, to update the plugin, click on the **OK** button in the left pane of Figure 2.39.

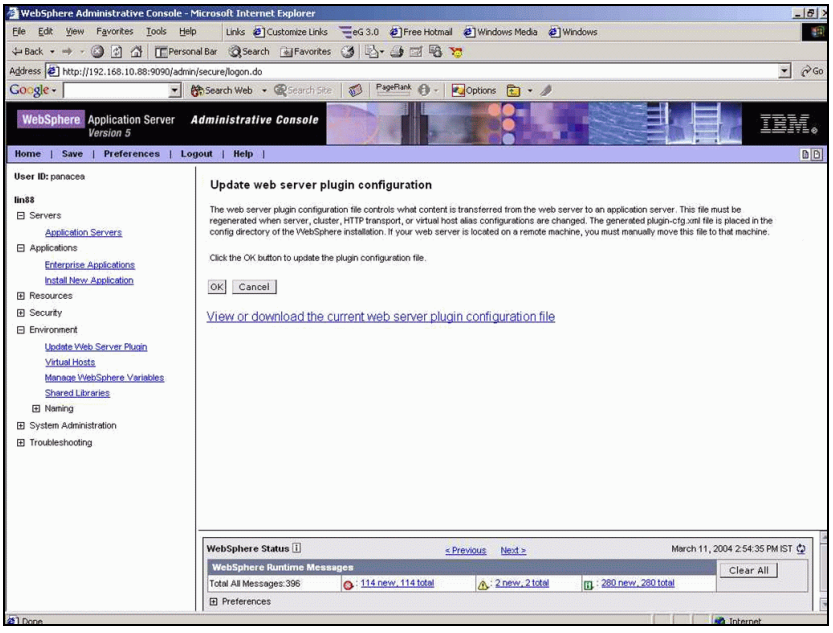

Figure 2.39: Clicking on the Update Web Server Plugin

19. If the web server plugin is updated successfully, the screen that appears next will display a

message to that effect (see Figure 2.40).

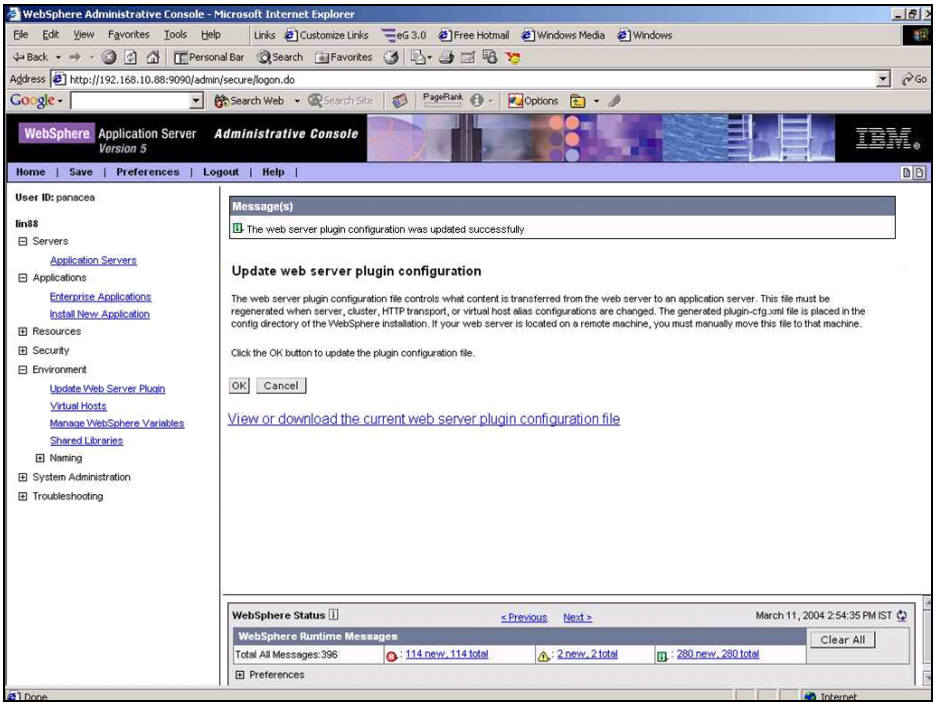

Figure 2.40: A page indicating that the web server plugin has been successfully updated

20. Finally, log out of the admin console and restart the WebSphere server. For that, first stop the server using the command **./stopServer.sh <servername>**, and then, start it using the command **./startServer.sh <servername>**. Both these commands are available in the **<WAS_HOME>/bin** directory, where **<WAS_HOME>** is the install directory of the WebSphere server.

2.5 Enabling Performance Monitoring

To enable performance monitoring for the WebSphere server 5.0, do the following:

1. Open WebSphere's admin console as explained in steps 1-3 of Section **2.4**.
2. In the screen that appears (see Figure 2.41), expand the **Servers** node in the left pane, and click on the **Application Servers** link within.

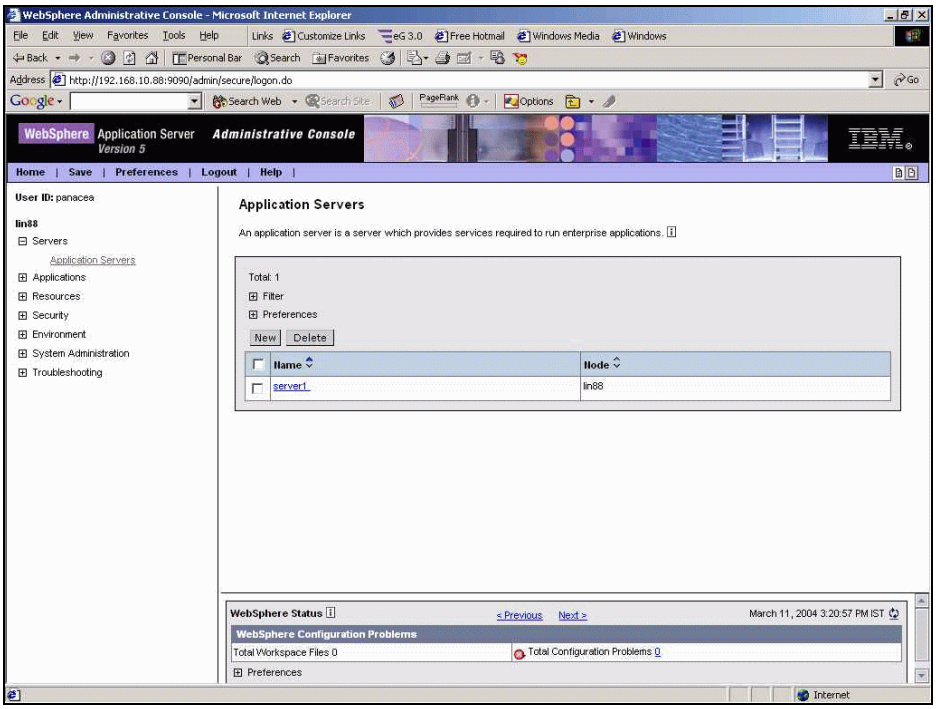

Figure 2.41: Viewing the list of application servers

3. As you can see, the right pane of Figure 2.41 displays the list of existing application servers. Now, click on the server that has been configured for monitoring by eG as indicated by Figure 2.41.
4. From the list of options displayed in the right pane of Figure 2.42, select the **Performance Monitoring Service** link.

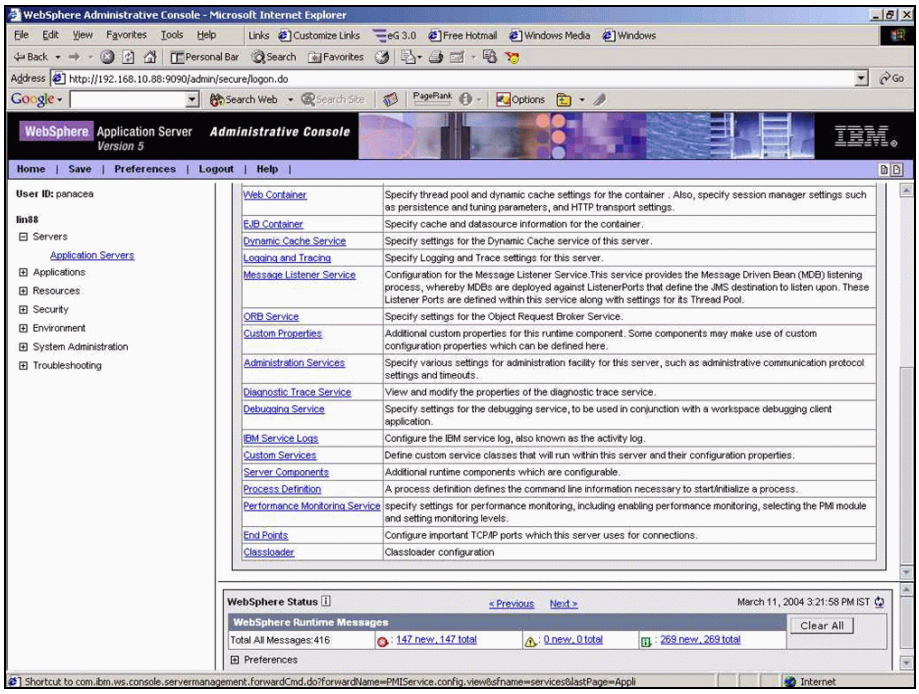

Figure 2.42: Selecting the Performance Monitoring Service option

5. In Figure 2.43 that appears, select the **Startup** check box, choose the **Custom** option, and set all the variables to X. Alternatively, the variables can also be set to H, but we recommend X. If you set the variables to H, then a few measures may not be available to you. Finally, click on the **Apply** button to register the changes.

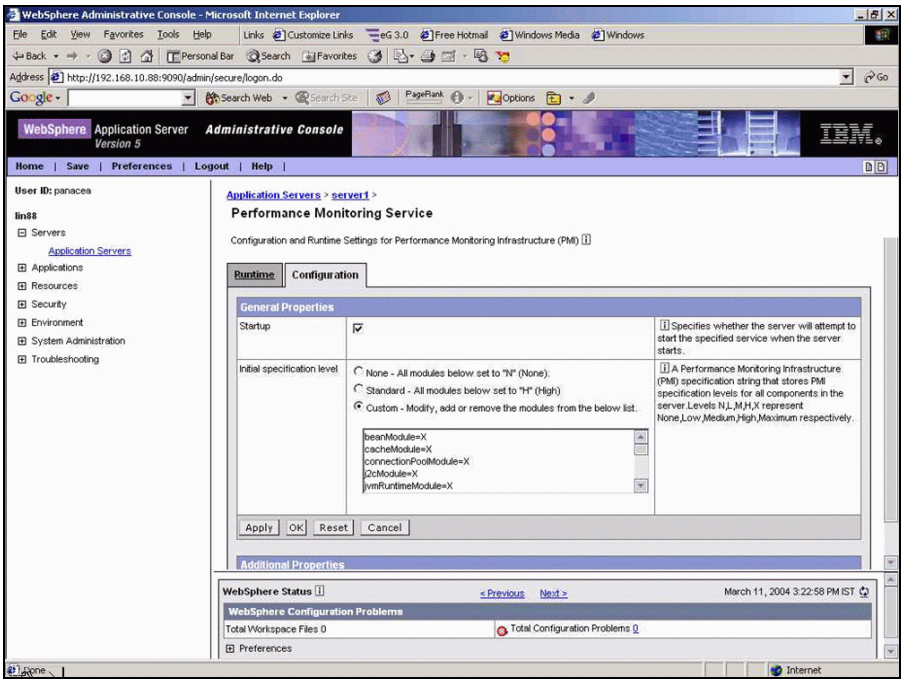

Figure 2.43: Enabling Performance Monitoring

2.6 Deploying the egurkha.ear file in a Clustered Environment

If the WebSphere server to be monitored is in a clustered environment, then the ear file deployment should be performed in the following manner:

1. Open Internet Explorer, and type the following URL: **http://<WebSphereIP>:<WebSpherePort>**, to connect to the administration console of the Node manager, which manages the application servers in the cluster.
2. Then, follow step 2 to step 8 of Section **2.4**.
3. Next, specify the application server in the cluster on which the **egurkha.ear** file is to be deployed, by selecting the cluster name from the **Clusters and Servers** list of Figure 2.44. Then, select the check box against the **egurkha** module in Figure 2.44, click the **Apply** button, and then the **Next** button.

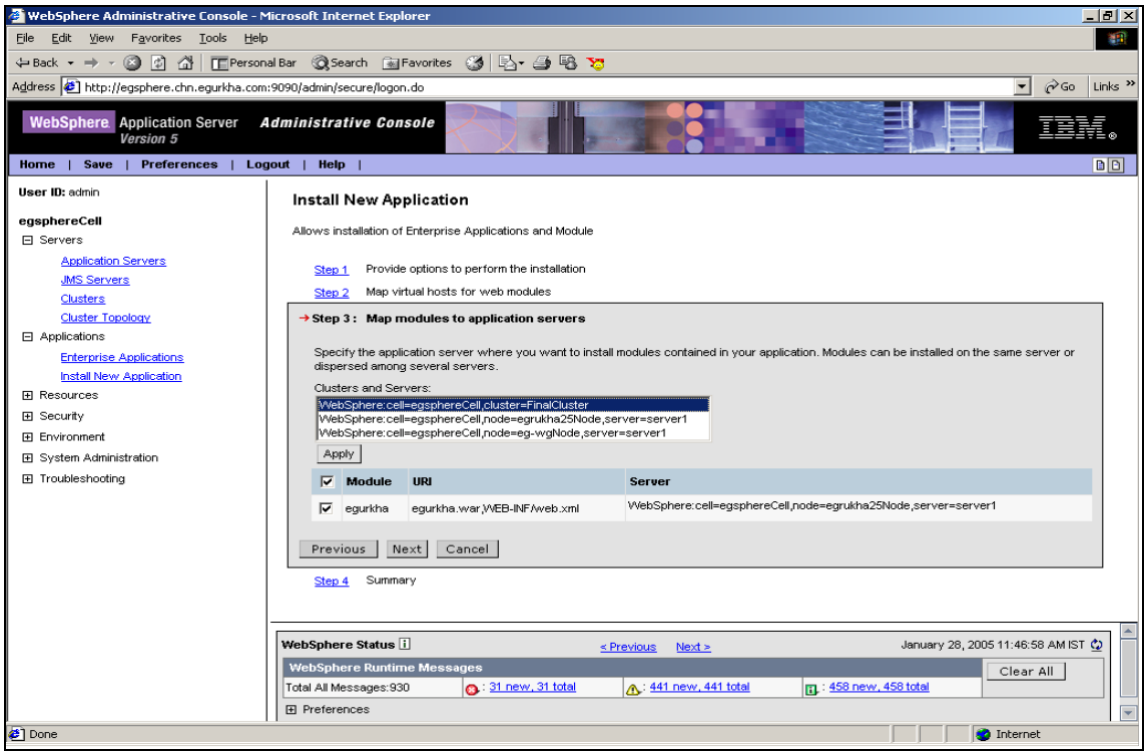

Figure 2.44: Mapping the newly installed application to a server in the cluster

4. Next, you will be prompted to map security roles to users/groups. If security has been enabled for the WebSphere server, then this mapping is mandatory. If not, simply click on the **Next** button to switch to the next step of the installation. In the case where security is enabled, it is recommended that you always map the *administrator* role with a specific *admin user/group* for monitoring purposes. To achieve this, follow step 10 to step 14 of the procedure discussed in Section **2.4**.
5. Figure 2.45 displaying a summary of the install options chosen will then appear.

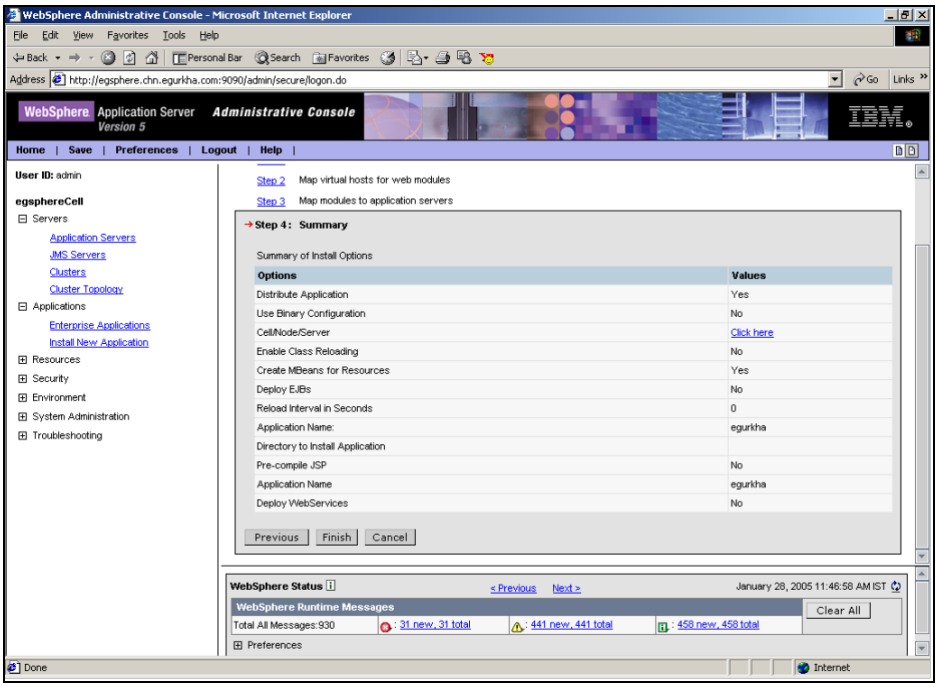

Figure 2.45: Summary of install options

6. Click on the **Finish** button in Figure 2.45 to begin installation of the **egurkha** application. Figure 2.46 will appear indicating the progress of the deployment.

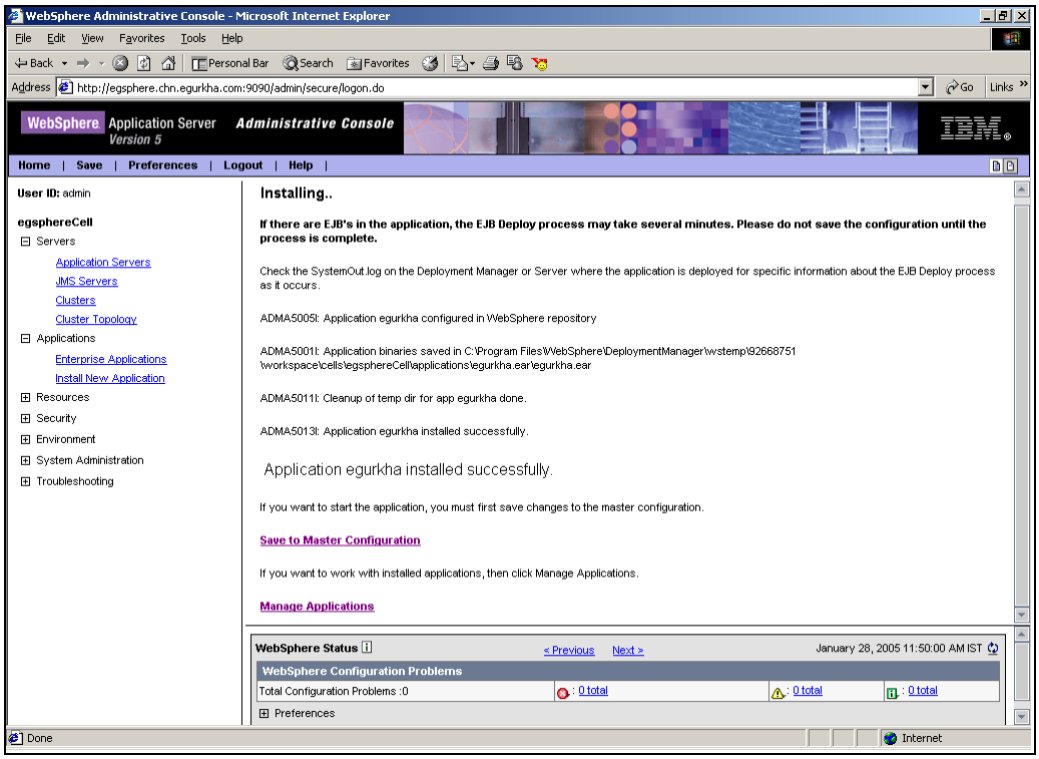

Figure 2.46: The progress of the ear deployment

7. Upon successful installation, a message to that effect will appear (see Figure 2.46). Now, click on the **Save to Master Configuration** link in Figure 2.46. Consequently, Figure 2.47 will appear wherein the **Save** button will have to be clicked in order to save the configuration changes to the node manager.

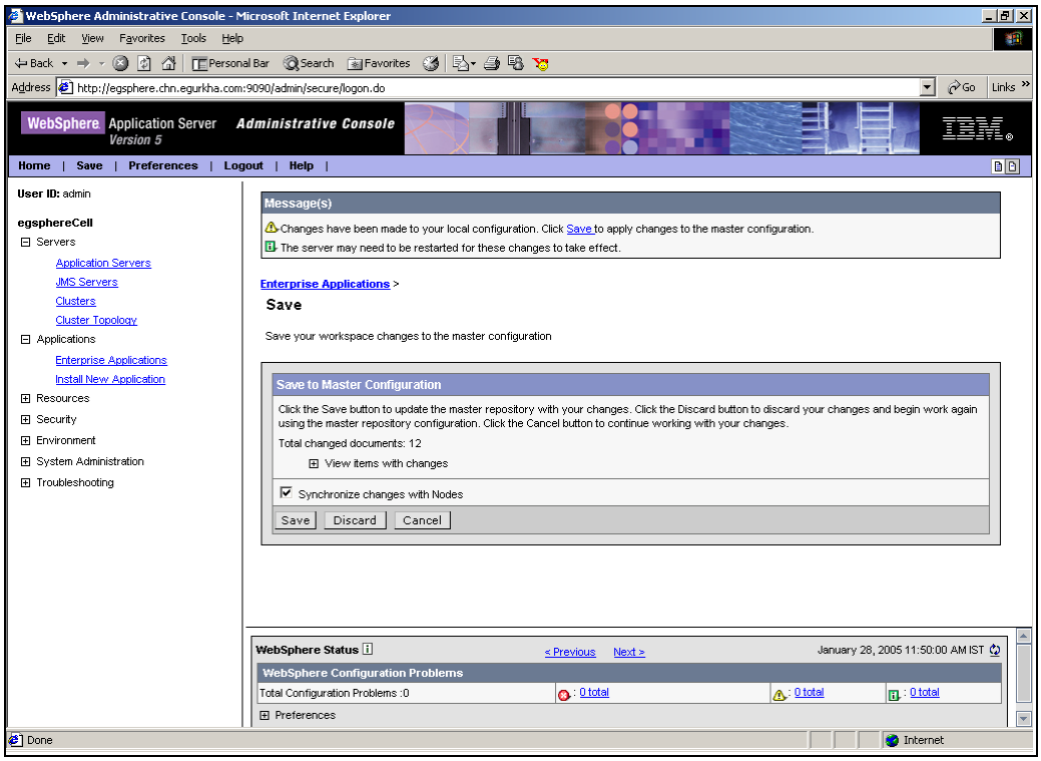

Figure 2.47: Saving the configuration changes

8. Finally, log out of the admin console and restart the WebSphere server. For that, first stop the server using the command **./stopServer.sh <servername>,** and then, start it using the command **./startServer.sh <servername>**. Both these commands are available in the **<WAS_HOME>/bin** directory, where **<WAS_HOME>** is the install directory of the WebSphere server.
9. Then, proceed to enable performance monitoring for the application servers to be monitored using the procedure discussed in Section **2.5**.

2.7 Verifying the Deployment of the 'egurkha' Application

To verify whether the *egurkha* application has been deployed properly or not, do the following:

- Open the Internet Explorer and specify the following URL in its Address bar: *http://<WebSphereIP>:<WebSpherePort>.*
- If the ear file has been deployed properly, then Figure 2.48 will appear:

Figure 2.48: The page that appears upon typing the specified URL in the Internet Explorer

- If the ear file has not been deployed, then an error message will appear instead.

2.8 Managing the IBM WebSphere Application – 4/5.x Server

After installation of the eG Agent, please follow the following steps to configure eG to monitor WebSphere Application from the eG Manager.

1. Login to the eG administrative interface as admin.
2. Check if the WebSphere application server is already discovered. If already discovered, use the **COMPONENTS - MANAGE/UNMANAGE** page that appears on using the Infrastructure -> Components-> Manage/Unmanage menu to manage it. Otherwise, change the port information if needed and run discovery to discover it or manually add the WebSphere Server. For adding, use the **COMPONENTS** page that can be accessed through the menu sequence Infrastructure -> Components -> Add/Modify. While components manually added are automatically managed, proceed to manually manage the discovered components using the **COMPONENTS - MANAGE/UNMANAGE** page.
3. Now, if you proceed to log out of the administrative interface, a list of unconfigured tests (see Figure 2.49) will appear prompting you to configure tests for the WebSphere Server.

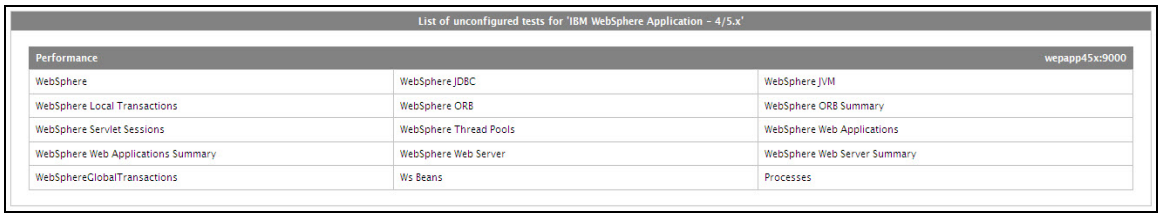

Figure 2.49: A page displaying WebSphere tests to be configured

4. Click on, say, **Ws Beans** test to move to the page that facilitates test configuration (see Figure 2.50). This test reports the performance metrics that determines the functioning of the EJB groups.

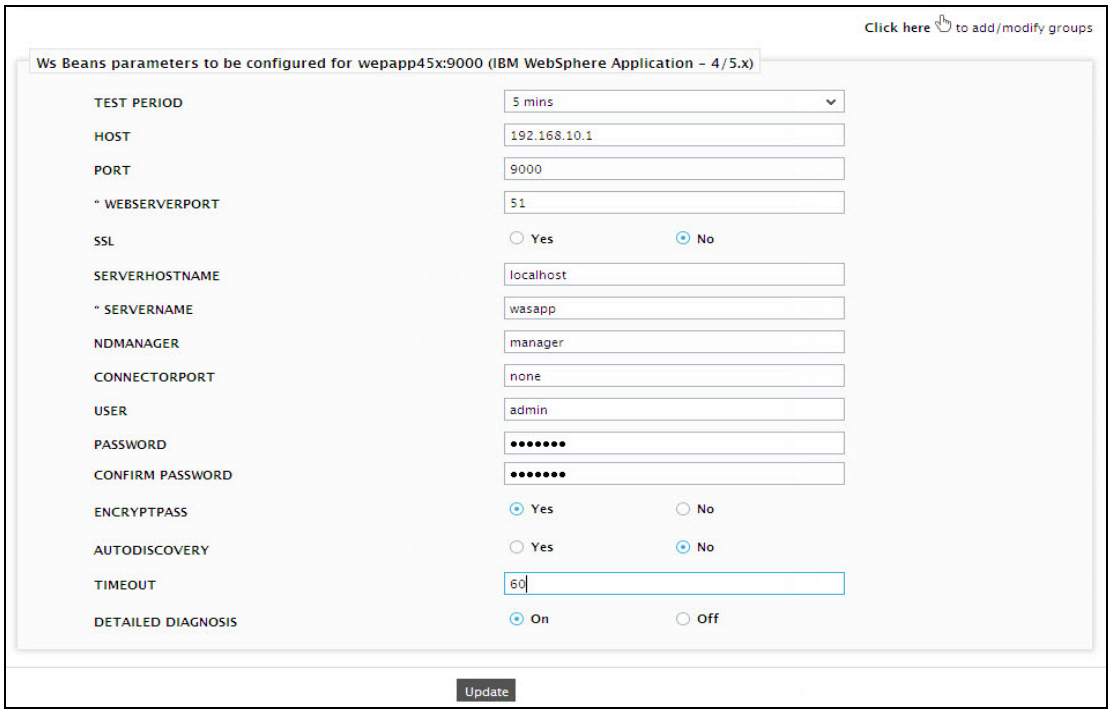

Figure 2.50: Specifying the test parameters for a WebSphere application

5. Here, select the WebSphere server for which the test is to be configured, click on the **Configure** button, and provide the following information:

   - **TEST PERIOD** - how often should the test be executed
   - **HOST** – The IP address of the WebSphere application server
   - **PORT** – The port number of the WebSphere application server

- **WEBSERVERPORT** – The port number through which the WebSphere applications can be accessed. (For IBMHTTPServer this information can be found in the **httpd.conf** file located in **$<INSTALLDIR>/conf** directory.)
- **SSL** - Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.
- **SERVERHOSTNAME** - Specify the node name of the server instance being monitored. To know the node name that is to be specified against the **SERVERHOSTNAME** parameter, do the following:
  - Login to the WebSphere Administrative Console.
  - Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within (see Figure 2.51).

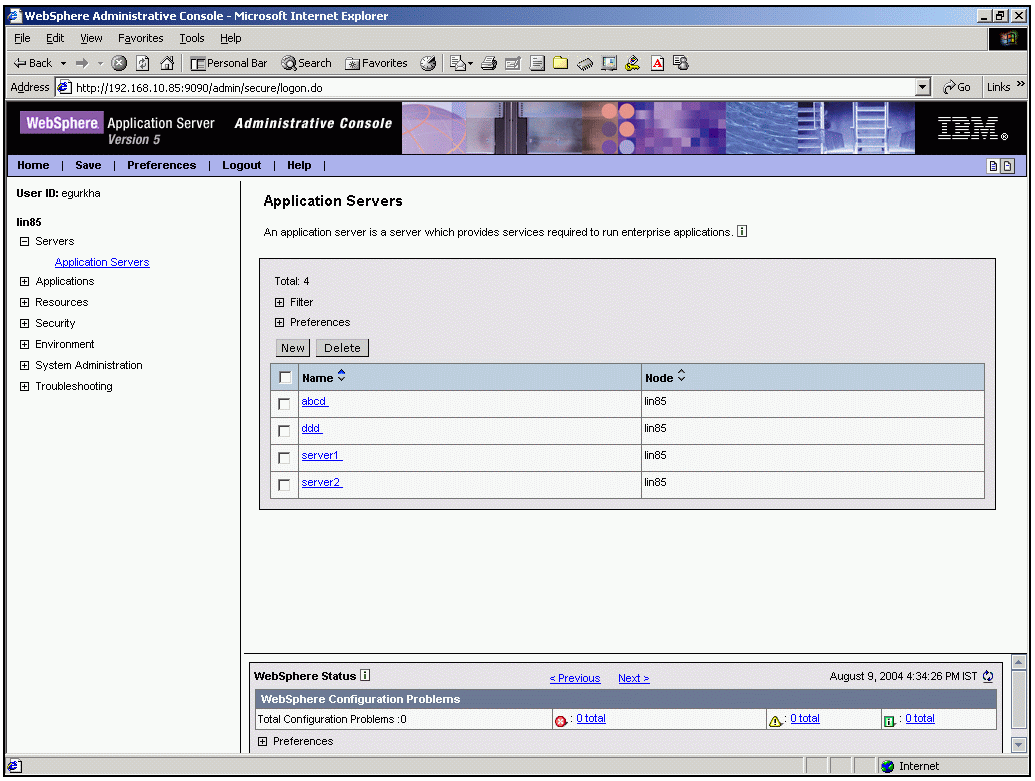

Figure 2.51: The WebSphere Administrative Console displaying the list of application server instances

- A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the node name that corresponds to the application server instance being monitored, and specify that name against the **SERVERHOSTNAME** parameter.

- **SERVERNAME** – Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, do the following:
  - Connect to the WebSphere Administrative console using the URL: http://<WebSphereIP>:<WebSpherePort>/, **of the WebSphere server:Port number of the WebSphere server>\admin**.
  - Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node (see Figure 2.52).

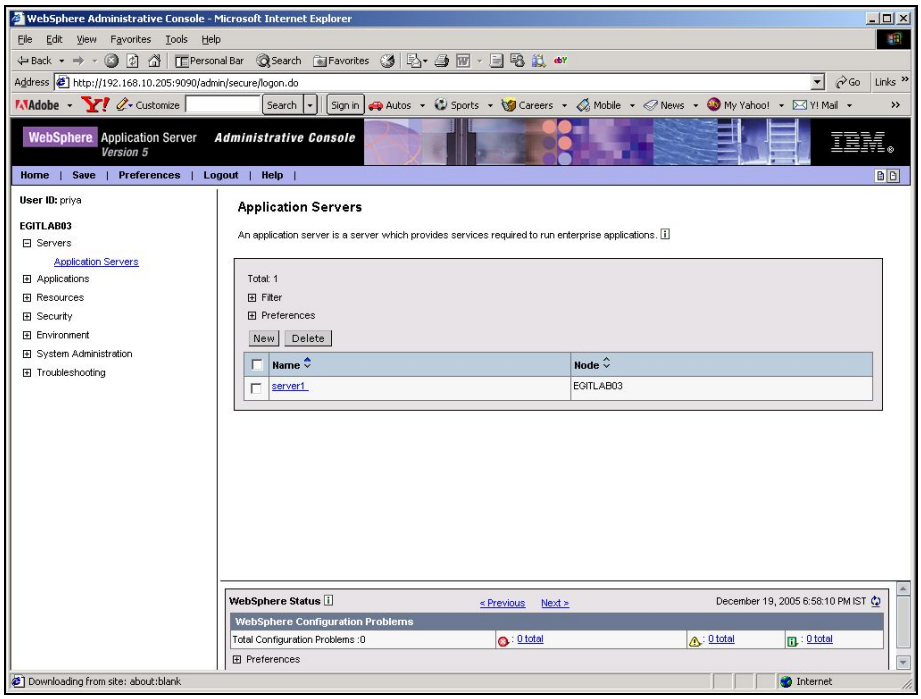

Figure 2.52: The instances of a WebSphere server

  - A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane (see Figure 2.52).
  - One of the displayed server instances can be specified as the value of the **SERVERNAME** parameter.

- **NDMANAGER** - The **NDMANAGER** parameter is applicable only under the following circumstances:

- If the WebSphere server being monitored belongs to a cluster, or,
- If the WebSphere server being monitored is one of many instances of the server running on the same host

In the case of situation (a), in the **NDMANAGER** text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the node manager, do the following:

- Login to the Administrative Console of the node manager using the URL *http://<WebSphereIP>:<WebSpherePort>*.
- Using the tree-structure in the left pane of the Administrative Console that appears (see Figure 2.53), drill down to the **Deployment Manager** node within **System Administration**.

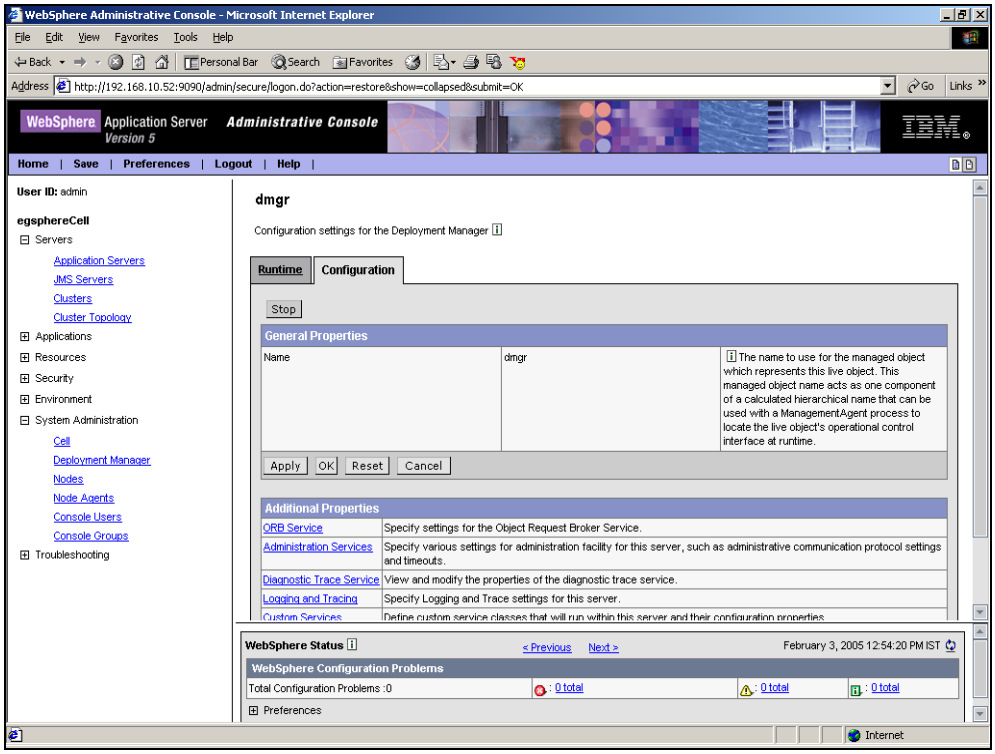

Figure 2.53: Opening the Deployment Manager node

- Select the **Configuration** tab that appears in the right pane (see Figure 2.53), and scroll down to the **End Points** link in the **Additional Properties** section.
- Once you locate the **End Points** link, click on it.
- Figure 2.54 will then appear. Here, select the **SOAP_CONNECTOR_ADDRESS** link.

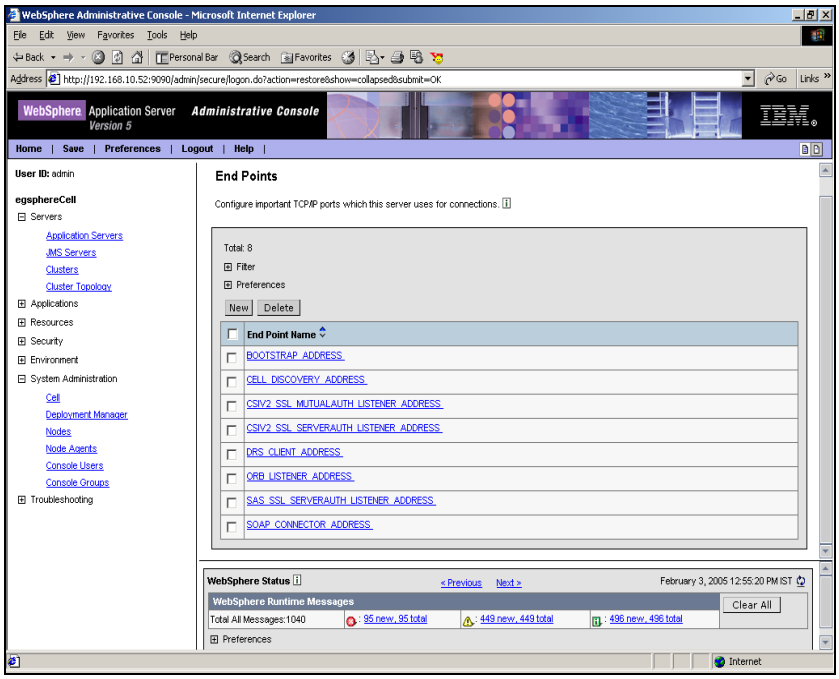

Figure 2.54: Clicking on the SOAP_CONNECTOR_ADDRESS link

- The fully qualified domain name displayed against **Host** in Figure 2.55 should be specified as the host name of the **NODE MANAGER** in the **NDMANAGER** text box of Figure 2.50.

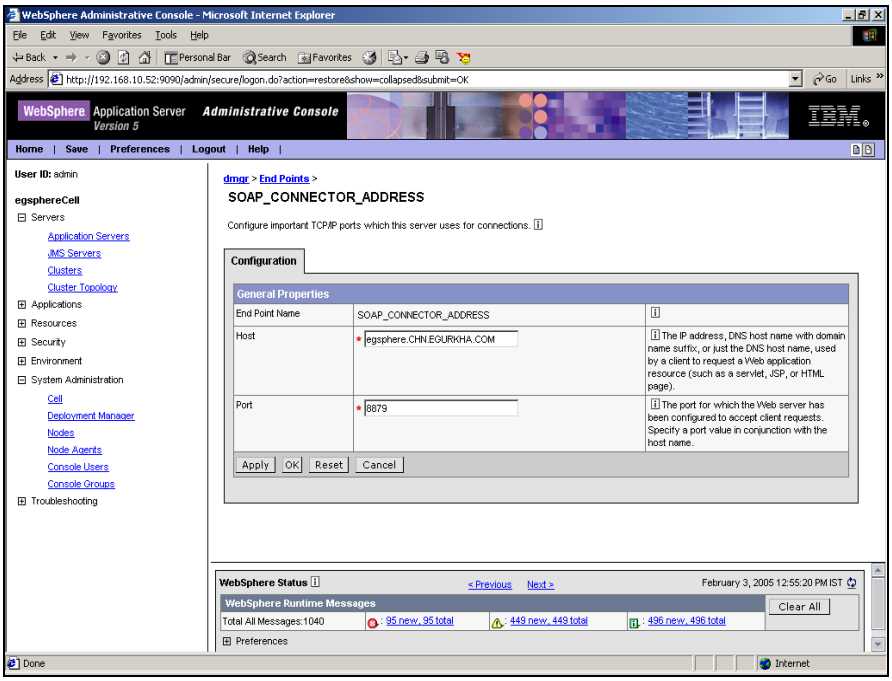

Figure 2.55: Viewing the host name of the node manager

In the case of situation (b), enter the **SERVERHOSTNAME** itself as the **NDMANAGER**.

If both conditions (a) and (b) do not apply, then specify none here.

- **CONNECTORPORT** – The **CONNECTORPORT** parameter is applicable only under the following circumstances:

  a. If the WebSphere server being monitored belongs to a cluster, or,

  b. If the WebSphere server being monitored is one of many instances of the server running on the same host

  In case of situation (a), the **CONNECTORPORT** parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the following:

  - From the Node manager's host, open the **<WEBSPHERE_ INSTALL_ DIR>\DeploymentManager\properties\wsadmin.properties** file (see Figure 2.56).

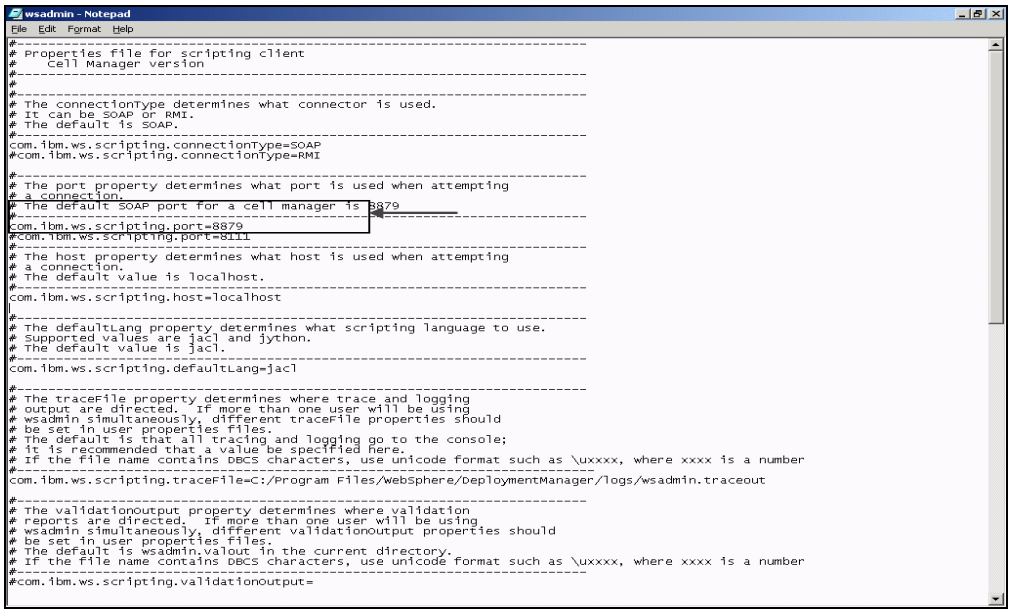

Figure 2.56: The wsadmin.properties file

- Figure 2.56 highlights two port-related entries in the file – one commented (using a #) and one uncommented. The uncommented entry denotes the active port, and its corresponding value (8879 in the example illustrated by Figure 2.56) will have to specified as the **CONNECTORPORT** in Figure 2.50.

In case of situation (b), do the following to know the connectorport.

- Login to the WebSphere Administrative Console.
- Expand the **Servers** node in the tree-structure in the left pane of the console, and click on the **Application Servers** link. A list of application server instances will then appear in the right pane (see Figure 2.57).

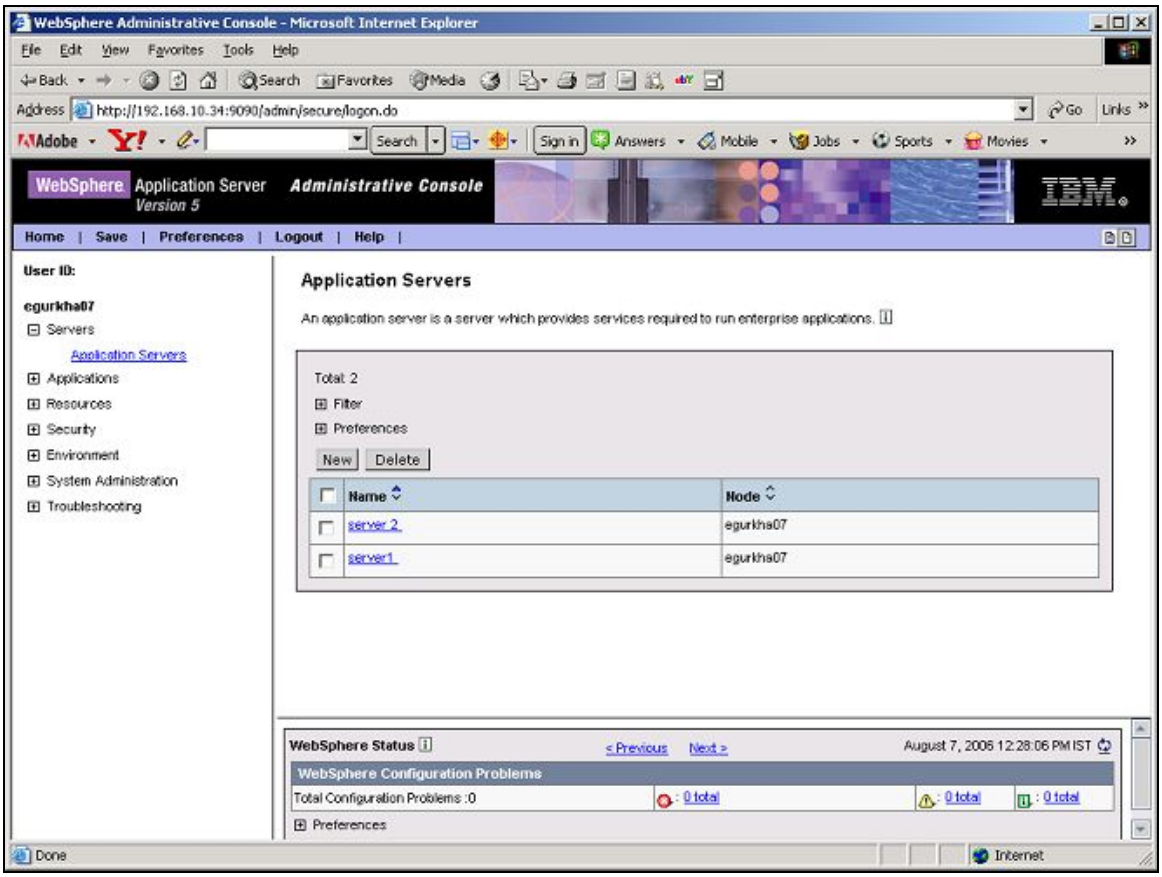

Figure 2.57: Opening the WebSphere Administrative Console

- Click on the server instance being monitored. Doing so invokes Figure 2.58 that displays the **Configuration** of the chosen server instance.

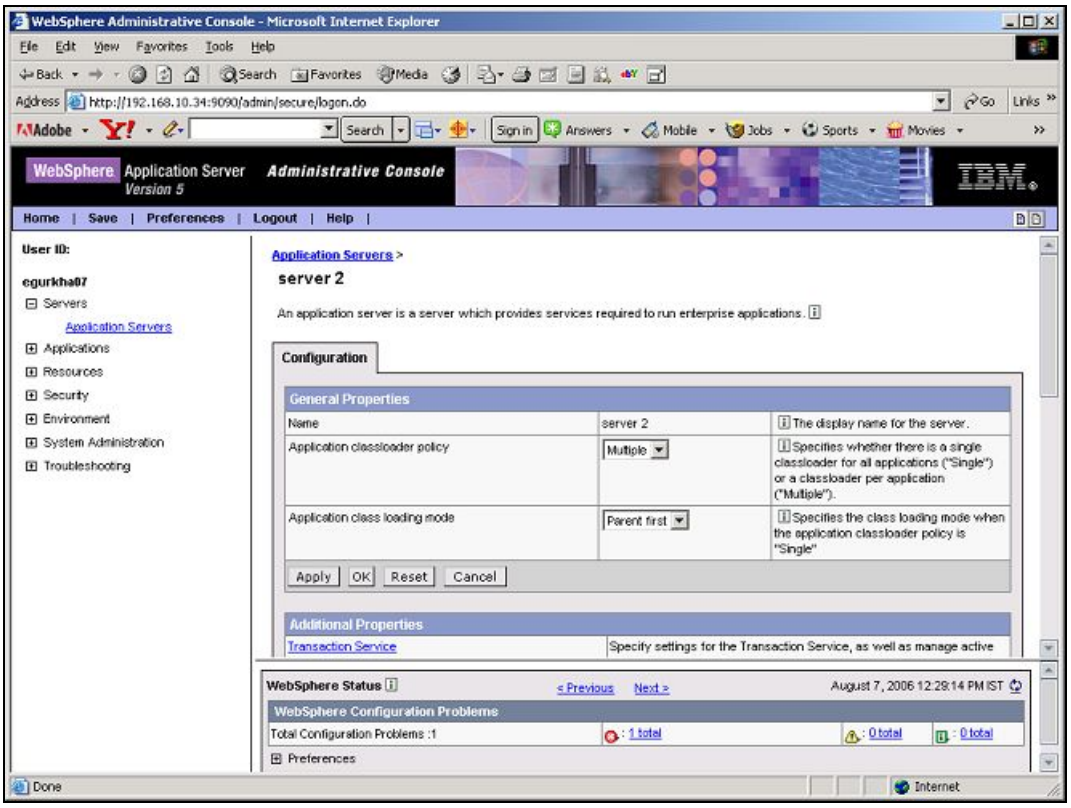

Figure 2.58: The Configuration of the server instance being monitored

- Scroll down the **Configuration** to view the **End Points** link (see Figure 2.59). Once you locate the **End Points** link, click on it.

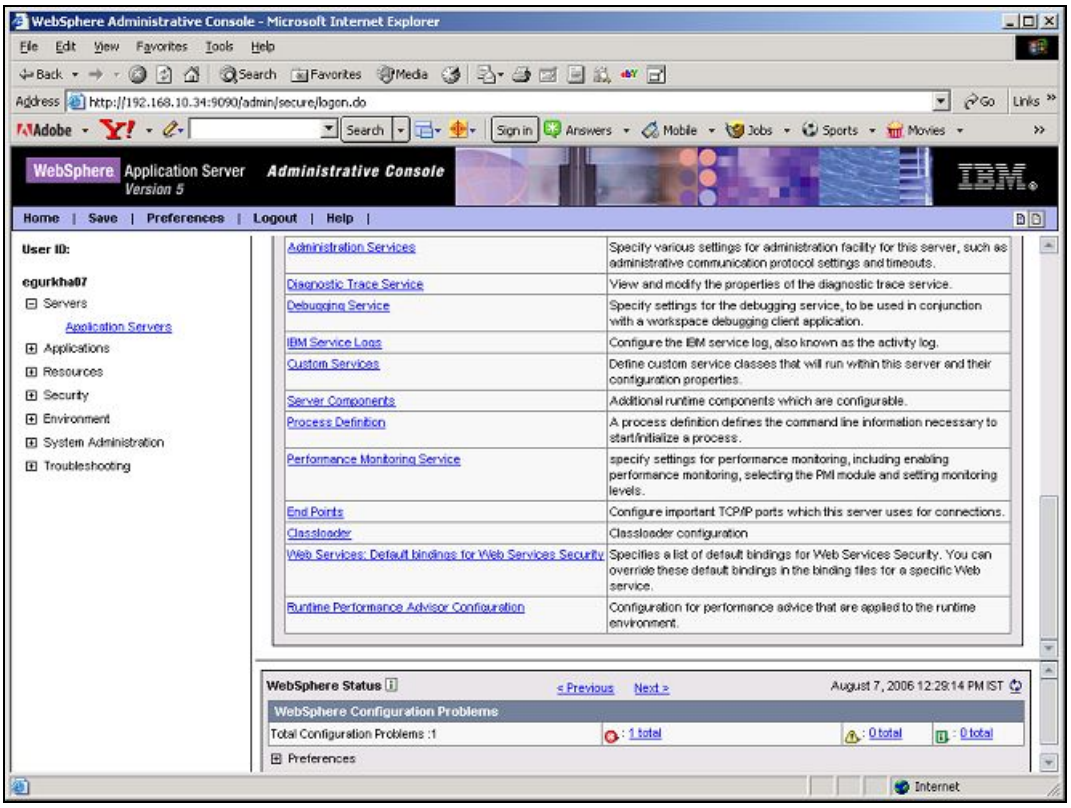

Figure 2.59: Clicking on the End Points link

- Figure 2.60 will then appear. Click on the **SOAP_CONNECTOR_ADDRESS** link in Figure 2.60.

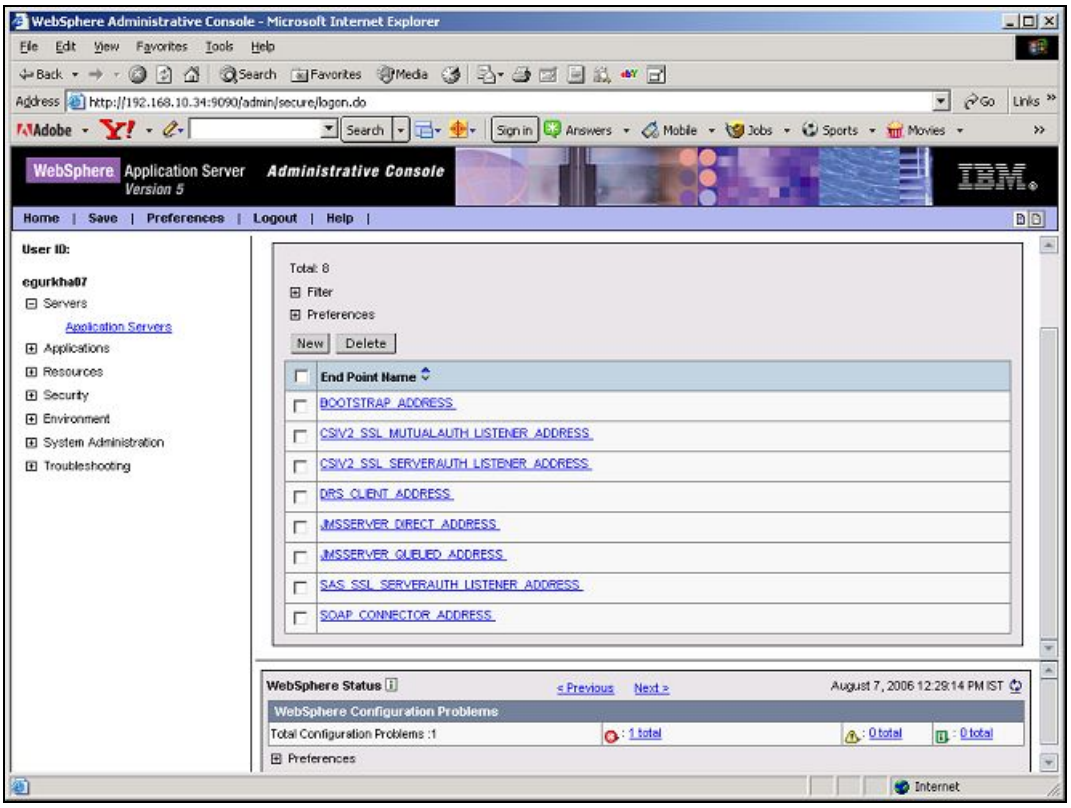

Figure 2.60: Clicking on the SOAP_CONNECTOR_ADDRESS link

- The **Port** displayed in Figure 2.61 that appears should be used as the **CONNECTORPORT** in situation (b).

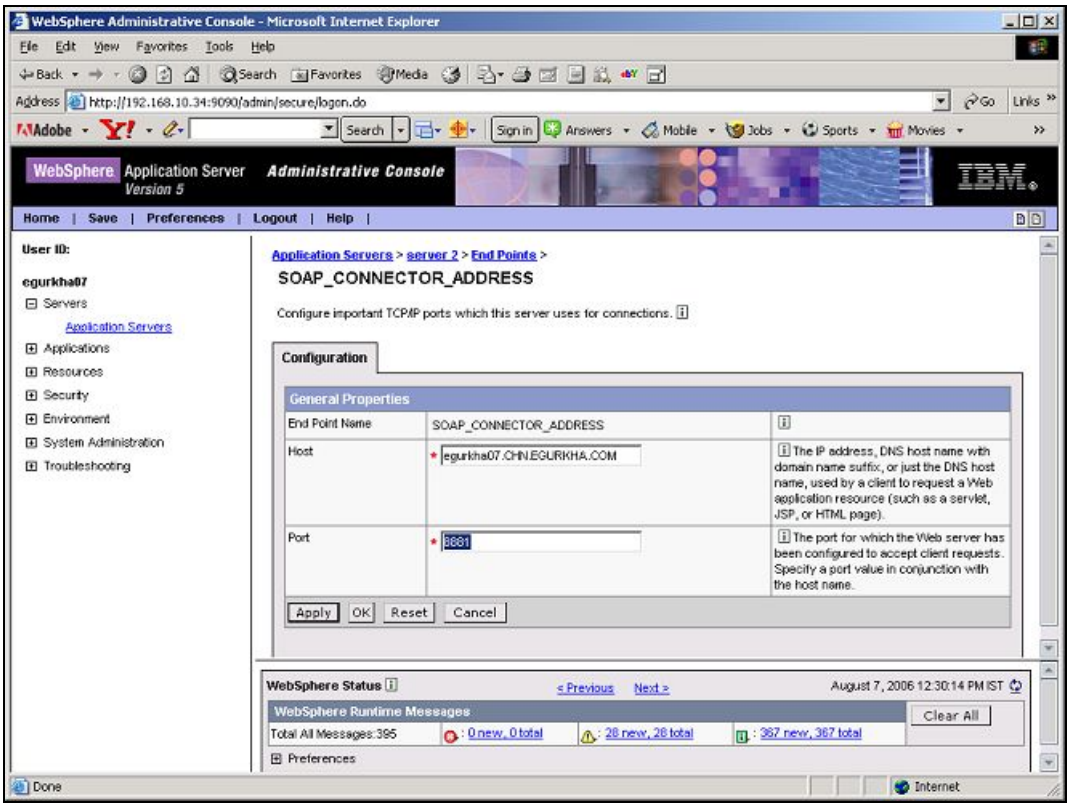

Figure 2.61: Viewing the CONNECTORPORT of the server instance being monitored

If both (a) and (b) do not apply, specify *none* against **CONNECTORPORT**.

- **USER** - If security has been enabled for the WebSphere server being monitored, then provide a valid **USER** name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the **USER** text box should contain the default value 'none'.
- **PASSWORD** – If security has been enabled for the WebSphere server being monitored, then provide the password that corresponds to the specified user name. If the WebSphere server does not require any authentication, then leave the **PASSWORD** text box with its default setting.
- **CONFIRM PASSWORD** - If security has been enabled, confirm the specified **PASSWORD** by retyping it in the **CONFIRM PASSWORD** text box. If the WebSphere server does not require any authentication, then leave the **CONFIRM PASSWORD** text box with its default setting.
- **ENCRYPTPASS** - By default, this flag is set to **yes**, indicating that the **PASSWORD** of the WebSphere server is encrypted by default. To disable password encryption, select the **no** option.

- **AUTODISCOVERY** – By default, the eG suite allows administrators to configure EJB groups using the eG administrative interface, and reports metrics pertaining to every group so created. Accordingly, by default, **AUTODISCOVERY** is set to **NO**. If you want EJBs to be discovered and monitored automatically, then select the **YES** option against **AUTODISCOVERY**. When this is done, the eG agent automatically discovers all the EJBs on the WebSphere server, and reports one set of measures for every EJB hosted on the server.
- **TIMEOUT** - In the **TIMEOUT** text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default **TIMEOUT** period is 60 seconds.
- **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the **DETAILED DIAGNOSIS** capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

  The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

  - The eG manager license should allow the detailed diagnosis capability
  - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

6. Then, click on the **Update** button to save the changes.
7. Now, wait for the test to run once, and then proceed to **Reconfigure** it for creating EJB groups. The eG agents will monitor only those EJB components of a WebSphere server that are part of an EJB group. To begin reconfiguring, click the **Reconfigure** button in the test configuration page. This results in the display of Figure 2.50. To configure the EJB groups, first click on the **Click here** hyperlink in Figure 2.50 and proceed in the same manner as discussed for the Ws Beans Test.
8. Once the EJB Groups are configured, try signing out again. This time it will prompt only for the process pattern. All other tests have shared the Web Server port information that we gave.
9. So, click on **Processes** test and proceed to configure it.

   Note:

   For **Processes** Test configuring procedure, refer to Configuring and Monitoring Mail Servers document.

10. Remember that the process pattern has to be specified in the following format:

    PROCESSNAME:PROCESSPATTERN

    For example, the process pattern can be specified as below:

    WebSphere:/opt/WebSphere/AppServer/java/jre/bin/exe/java*- Xmx128m*- Xminf0.15*- Xmaxf0.25*–X**

11. Use the context sensitive eG help to understand more about how this process pattern is coined.
12. Upon completion of Processes test configuration, signout of the administrative interface.
13. The WebSphere Application server is now ready to be monitored by eG.

Chapter 3: How does eG Enterprise Monitor IBM WebSphere Application Server (6.0 and above)?

In the case of the WebSphere application server version 6.0 and above, the eG agent uses the JMX architecture supported by the WebSphere server to gather the required metrics.

This section explains the steps involved in configuring and monitoring a WebSphere Application Server 6.0 (and above)

3.1 Configuring a WebSphere Application Server 6.x to Work with the eG Agent

To monitor a WebSphere Application server 6.x (and above), a specific WebSphere monitoring eG component has to be installed on it. If the WebSphere server to be monitored uses a SOAP connector port for its internal communications, then the **egurkha6.ear** component will have to be installed on the server. On the other hand, if the target WebSphere server uses an RMI port for communicating internally, then the **egurkha6rmi.ear** component should be installed on the server. Before installing either of these components on a WebSphere server, do the following:

- Copy the **egurkha6.ear** (or **egurkha6rmi.ear**) file from the <EG_INSTALL_DIR>\lib directoy (on Windows; on Unix, this will be **/opt/egurkha/lib**) to any temporary location on the eG agent host.
- Rename the file in the temporary folder as **egurkha.ear**.
- Copy the **egurkha.ear** file back to the <EG_INSTALL_DIR>\lib directoy (on Windows; on Unix, this will be **/opt/egurkha/lib**).

Then, follow the steps given below to perform the installation.

1. Login to the WebSphere Administrative Console by first accessing the URL: http://<IP/hostname of the WebSphere server>:<WebSpherePort>/, and then providing a valid **User ID** and **Password**.

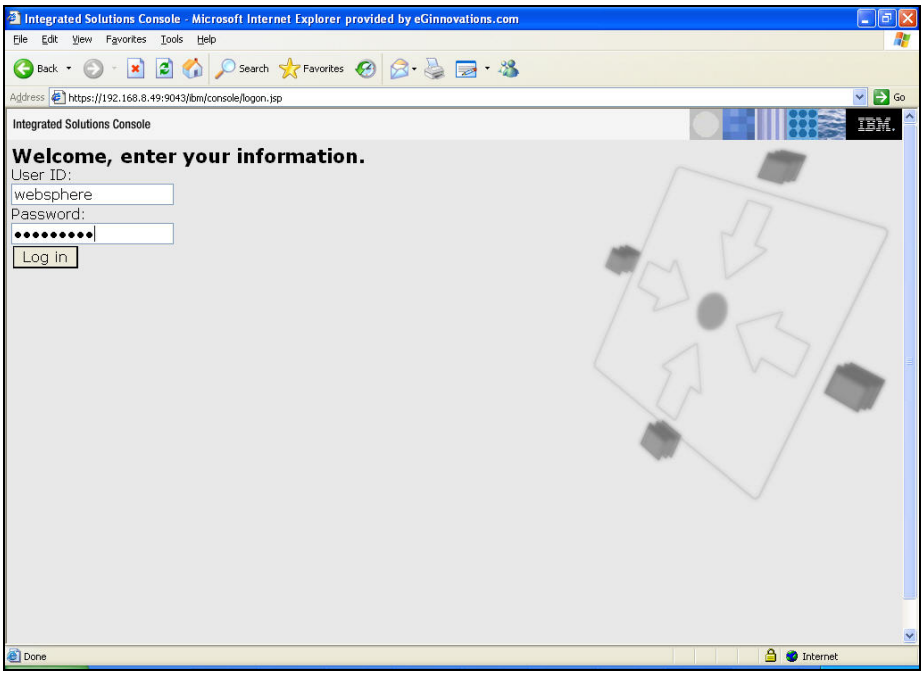

Figure 3.1: Logging into the WebSphere administrative console

2. The **WebSphere Administrative Console** will then appear as depicted by Figure 3.2.

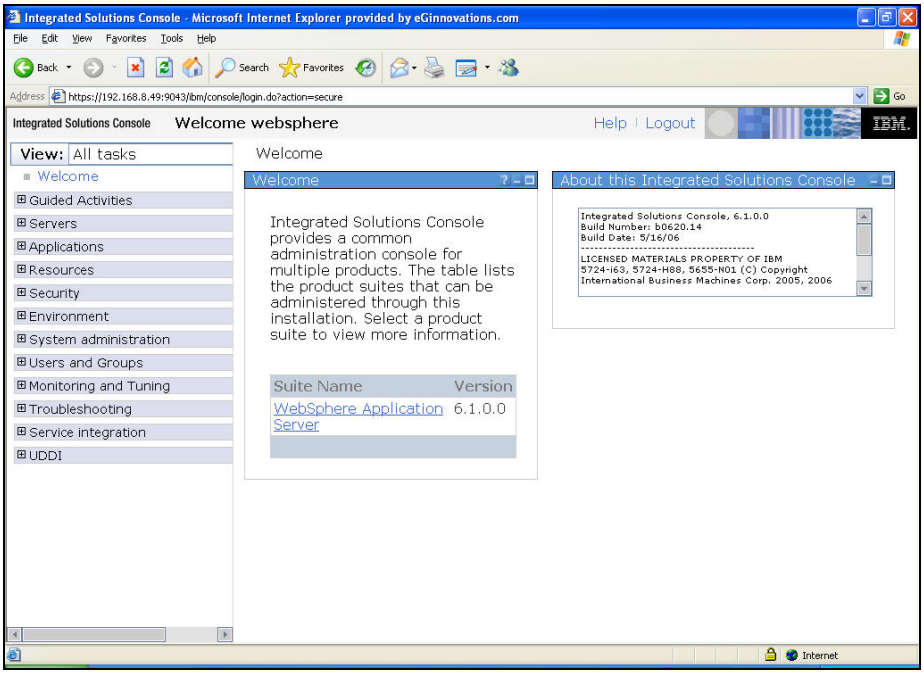

Figure 3.2: The WebSphere Administrative Console

3. Expand the **Applications** node in the tree-structure in the left pane of Figure 3.2, and then click

on the **Install New Application** sub-node within.

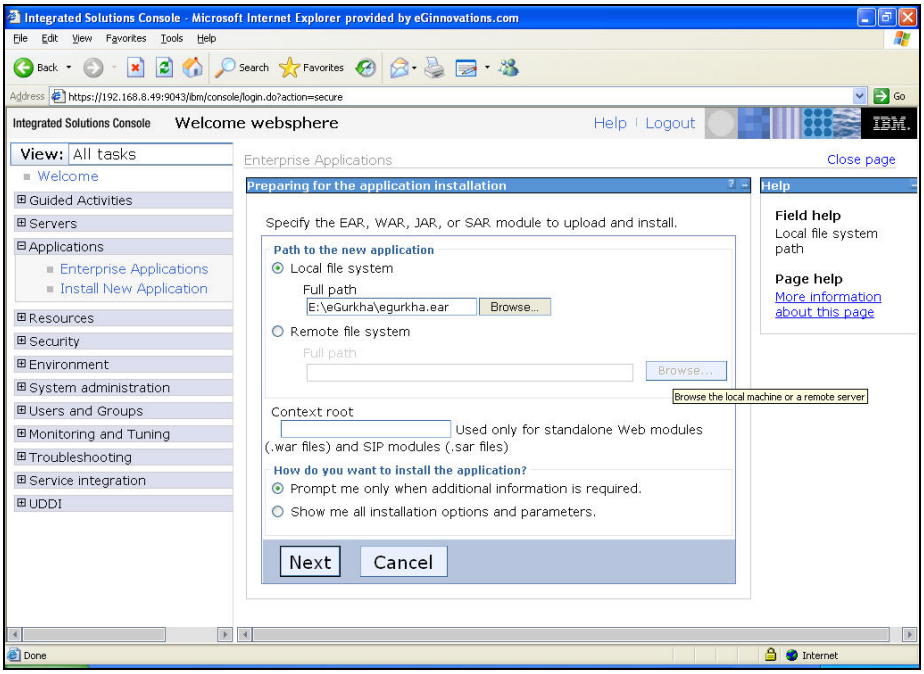

Figure 3.3: Installing a new application

4. Figure 3.3 will then appear. If the **egurkha.ear** file to be deployed is on the local host, then, select the **Local file system** option, and then specify the full path to the **egurkha.ear** application in the **Full path** text box. On the other hand, if the **egurkha.ear** file is available on a remote host in the environment, then, pick the **Remote file system** option and specify the full path to the remote **egurkha.ear** file in the **Full path** text box below. Typically, the 'ear' file will be available in the **/opt/egurkha/lib** directory (on Unix; on Windows, it will be the <EG_INSTALL_DIR>\lib directory). Then, click the **Next** button in Figure 3.3.

5. Using Figure 3.4 that appears next, view the **Installation Options**, and select the options that you want to enable for your application deployment. Since the default installation settings apply to the **egurkha.ear** deployment, you need not explicitly enable/disable any specific options in this page. Therefore, simply scroll down the page and click the **Next** button therein (see Figure 3.5).

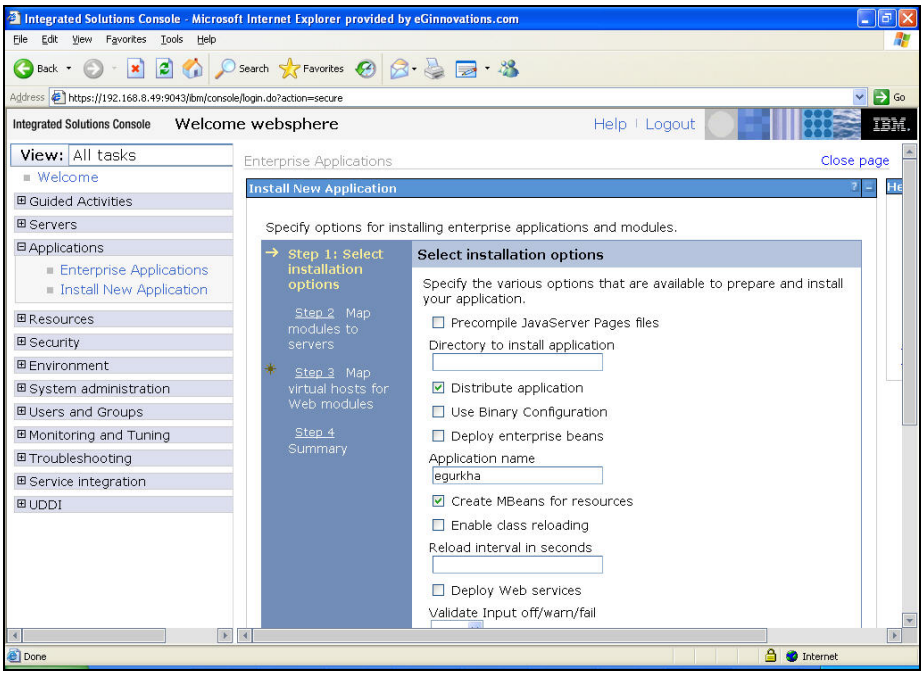

Figure 3.4: Viewing the installation options

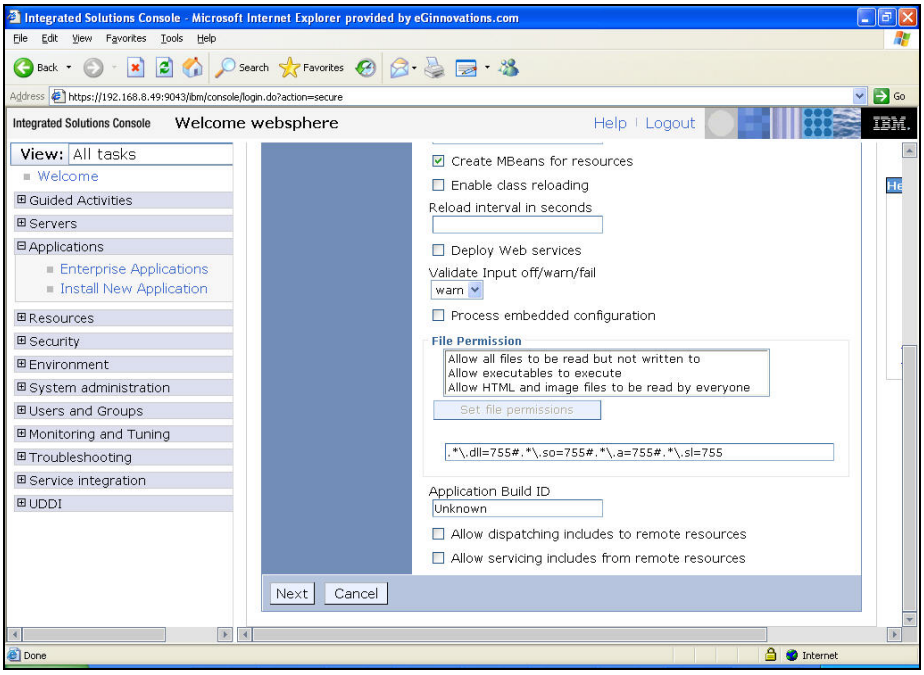

Figure 3.5: Scrolling down the installation options and clicking the Next button

6. When Figure 3.6 appears, map the **egurkha** module to the displayed **Server**, by selecting the check box corresponding to the **egurkha** module and clicking the **Next** button.

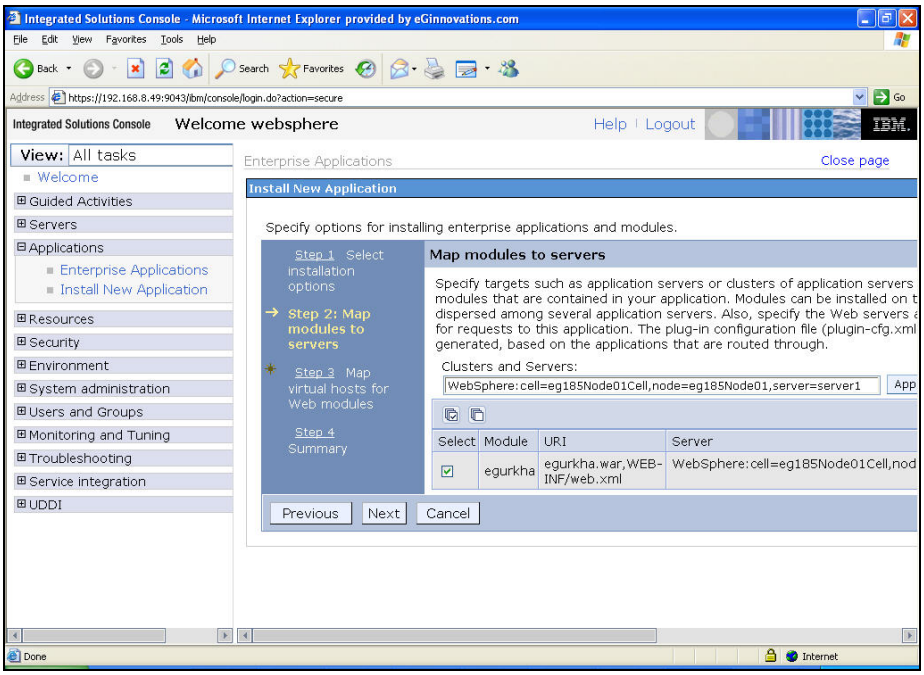

Figure 3.6: Mapping the egurkha module with a server

7. Then, to map the **egurkha** web module to the default selection in the **Virtual Host** drop-down, select the check box corresponding to the **egurkha** module in Figure 3.7, and click the **Next** button. This will ensure that the **egurkha** web module is installed on the default virtual host.

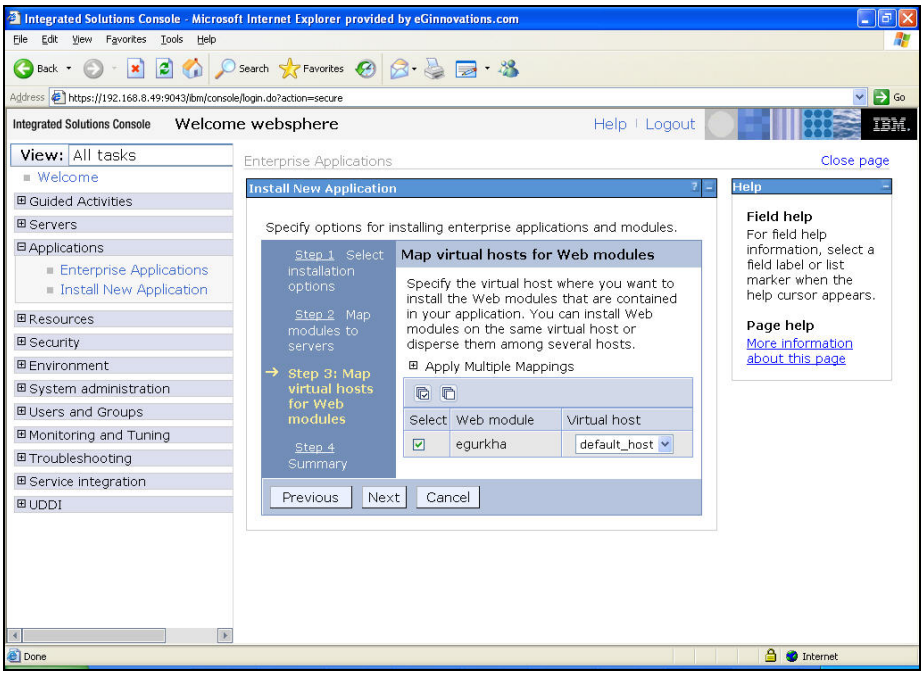

Figure 3.7: Mapping the egurkha web module with a virtual host

8. Next, a summary of your specifications for **egurkha.ear** deployment will appear (see Figure 3.8). Scroll down the summary to click the **Finish** button.

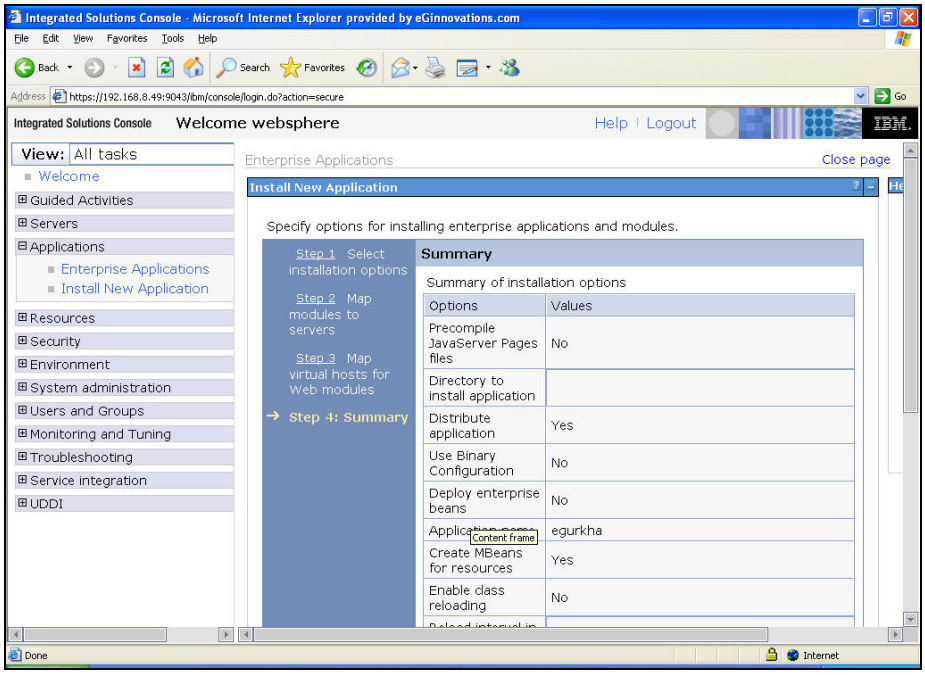

Figure 3.8: Viewing a summary of your installation specifications

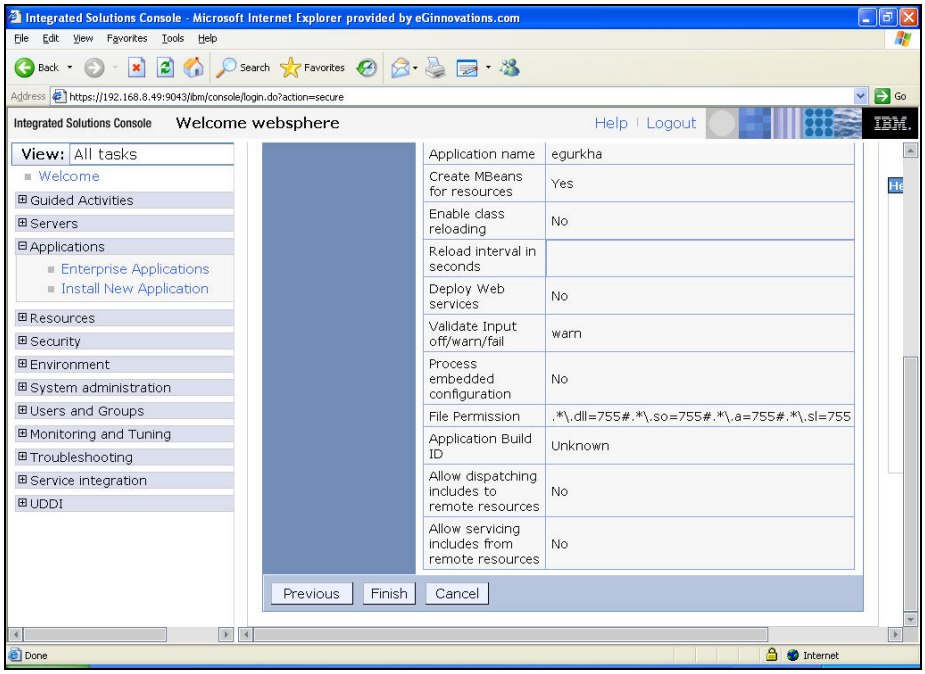

Figure 3.9: Scrolling down the summary and clicking the Next button

9. This will trigger the deployment of the **egurkha** application. Figure 3.10 indicates the status of the installation.

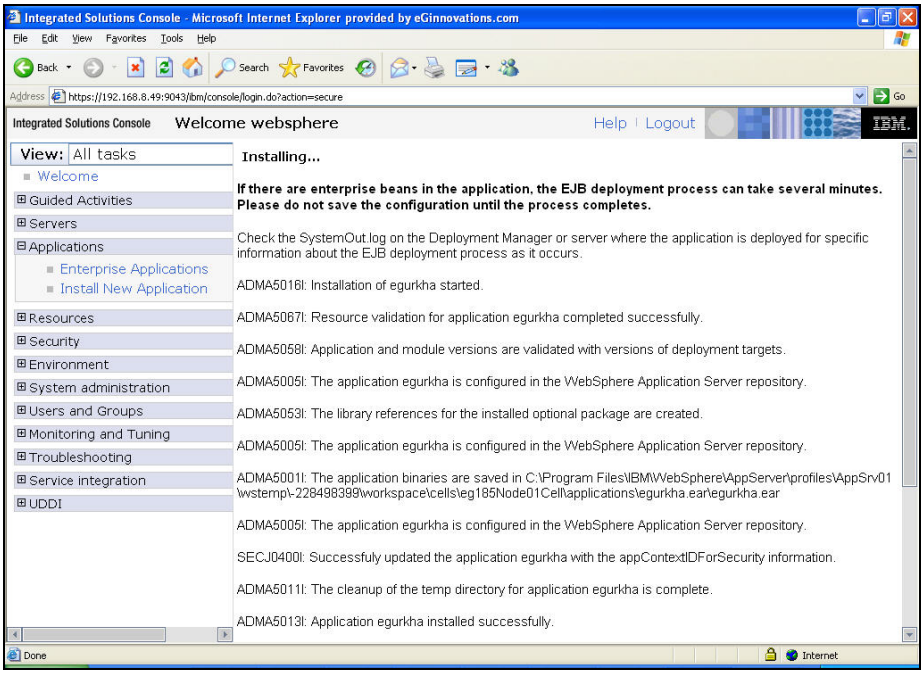

Figure 3.10: Status of the egurkha installation

10. Scroll down Figure 3.10 to determine whether the **egurkha** application was successfully deployed or not. If application deployment is successful, a message to that effect will appear here (see Figure 3.11).

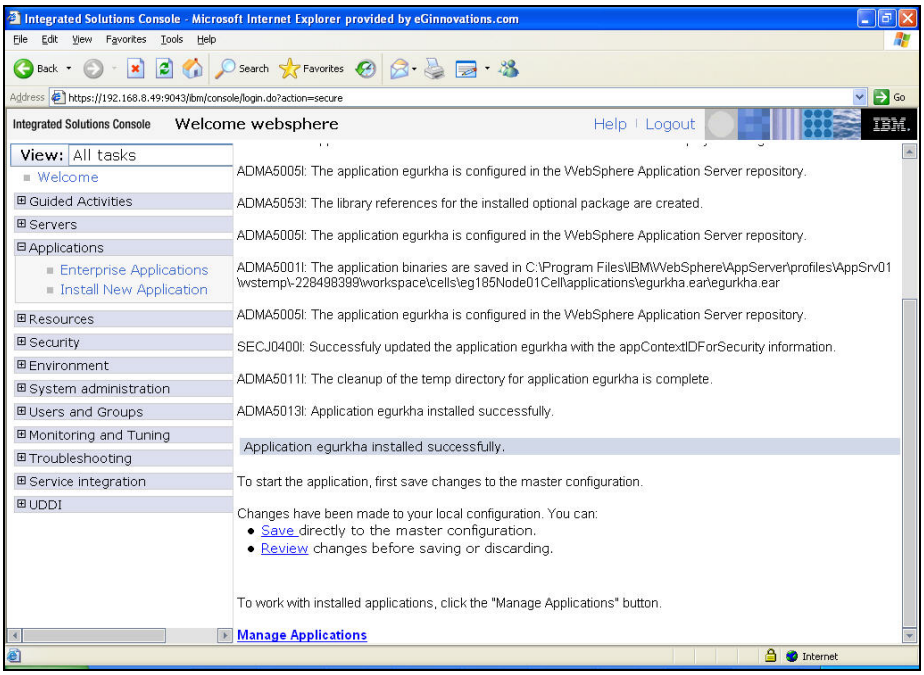

Figure 3.11: A message indicating that the egurkha application has been successfully installed

11. Then, proceed to save the **egurkha** application. The first step towards this is to save the changes to the master configuration. Before saving the master configuration, you may want to review the changes effected on it. For this, click the **Review** link in Figure 3.11.
12. Figure 3.12 will then appear displaying the **Total** number of **changed documents**. To view the list of documents that changed, expand the **Total changed documents** node (see Figure 3.13).

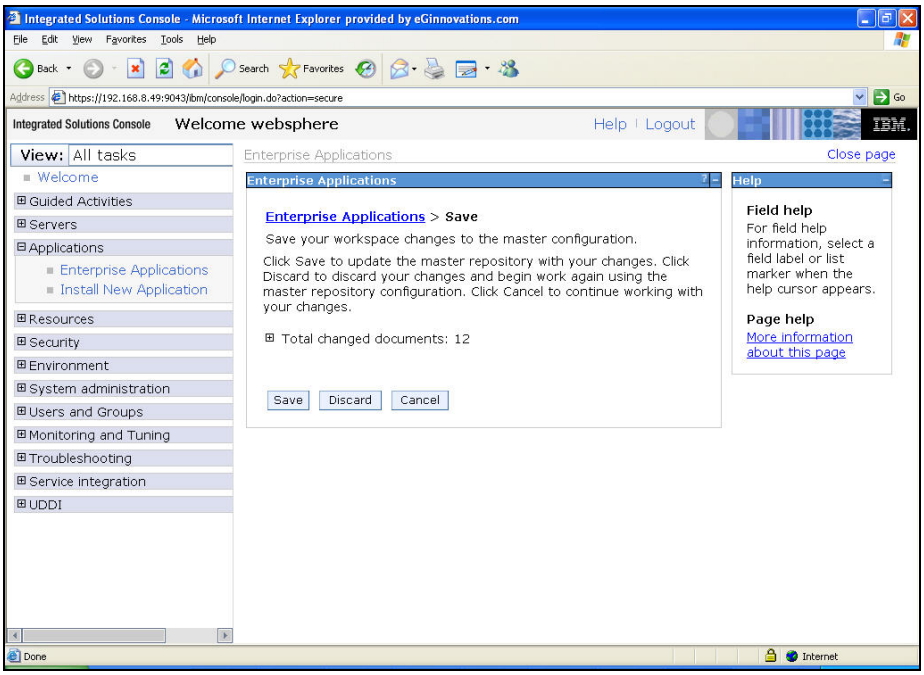

Figure 3.12: Viewing the total number of changed documents

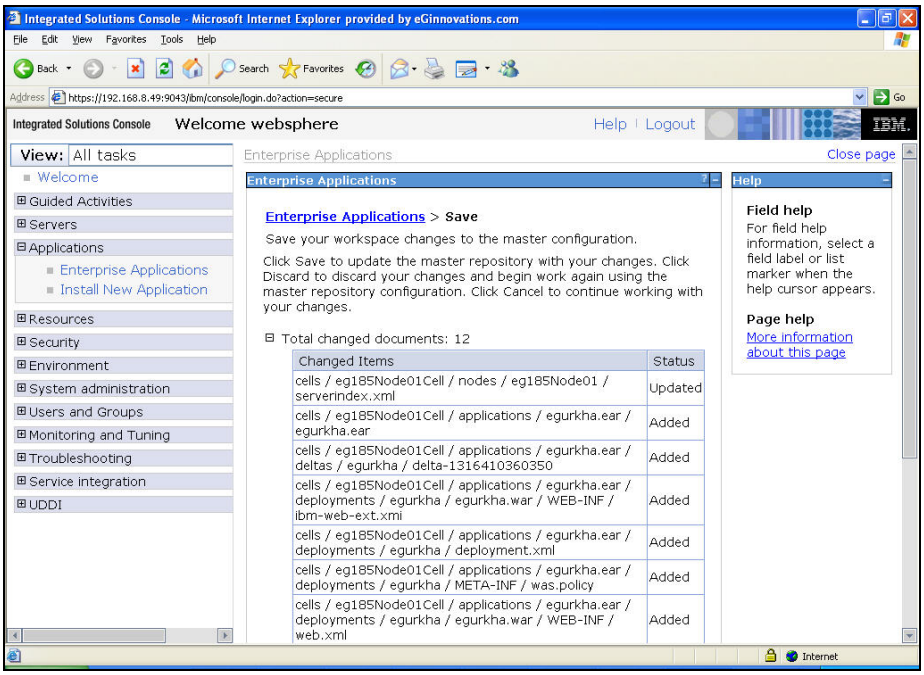

Figure 3.13: Reviewing the changes to the documents

13. Scroll down Figure 3.13 to review the complete list of changes. Finally, click the **Save** button in Figure 3.14 to save the changes.

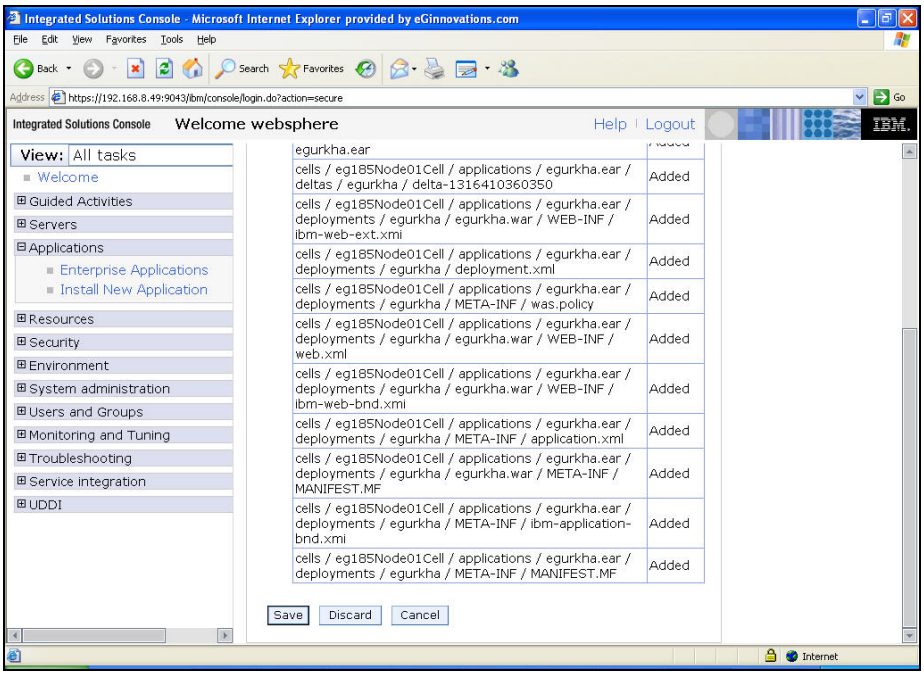

Figure 3.14: Saving the changes

14. Now, proceed to start the **egurkha** application. For that, first, click on the **Enterprise Applications** node in the tree-structure in the left panel of Figure 3.14. The list of installed applications will then appear in the right panel. Select the check box corresponding to the **egurkha** application and click the **Start** button therein to start it (see Figure 3.15).

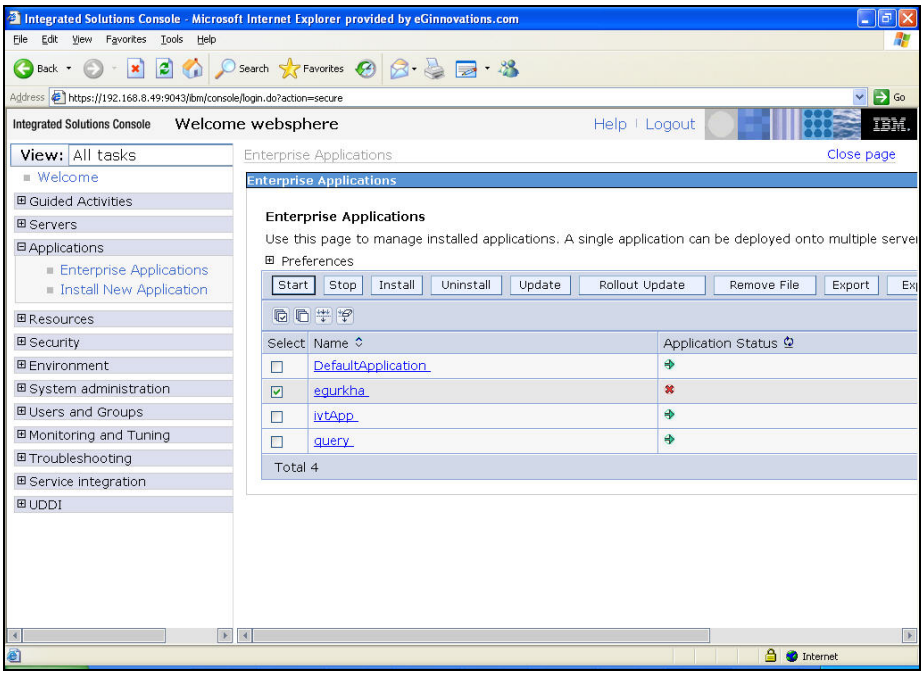

Figure 3.15: Selecting the Save to Master Configuration link

15. Once the application is started, a message to that effect will appear as depicted by Figure 3.16 below.

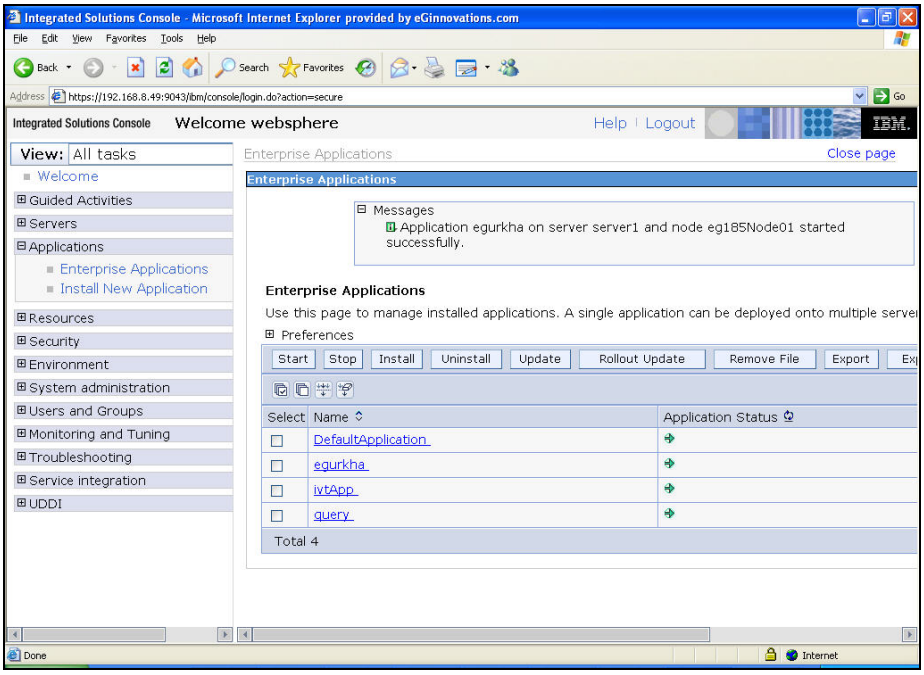

Figure 3.16: A message indicating that the egurkha application has started successfully

16. Finally, log out of the admin console.

3.2 Configuring a WebSphere Application Server 7.x to Work with the eG Agent

To monitor a WebSphere Application server 6.x (and above), a specific WebSphere monitoring eG component has to be installed on it. If the WebSphere server to be monitored uses a SOAP connector port for its internal communications, then the **egurkha6.ear** component will have to be installed on the server. On the other hand, if the target WebSphere server uses an RMI port for communicating internally, then the **egurkha6rmi.ear** component should be installed on the server. Before installing either of these components on a WebSphere server, do the following:

- Copy the **egurkha6.ear** (or **egurkha6rmi.ear**) file from the <EG_INSTALL_DIR>\lib directoy (on Windows; on Unix, this will be **/opt/egurkha/lib**) to any temporary location on the eG agent host.
- Rename the file in the temporary folder as **egurkha.ear**.
- Copy the **egurkha.ear** file back to the <EG_INSTALL_DIR>\lib directoy (on Windows; on Unix, this will be **/opt/egurkha/lib**).

Then, follow the steps given below to perform the installation.

1. Login to the WebSphere Administrative Console by first accessing the URL: http://<IP/hostname of the WebSphere server>:<WebSpherePort>/, and then providing a valid **User ID** and **Password**.

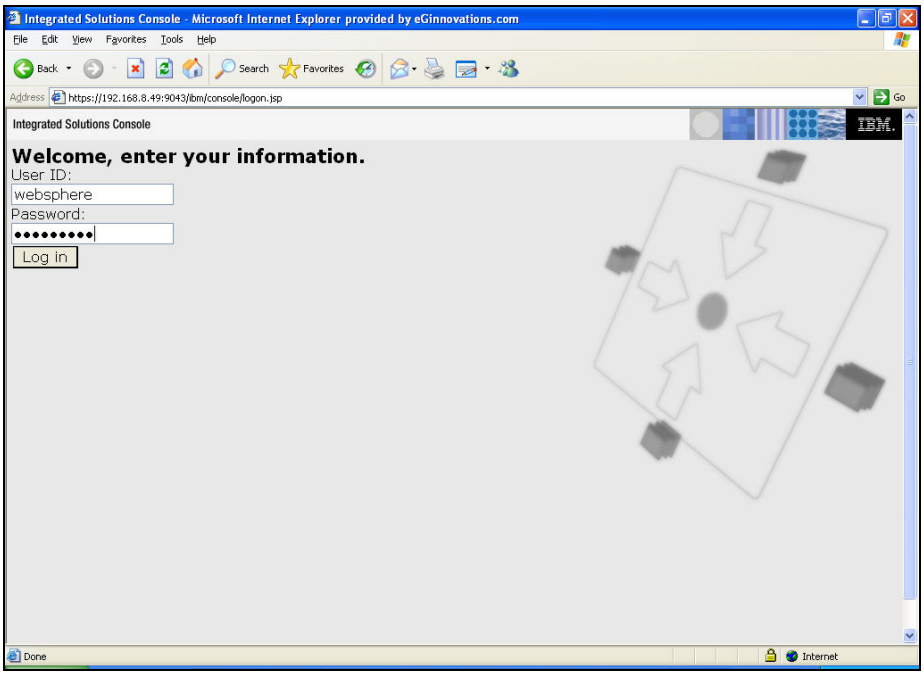

Figure 3.17: Logging into the WebSphere administrative console

2. The **WebSphere Administrative Console** will then appear as depicted by Figure 3.18.

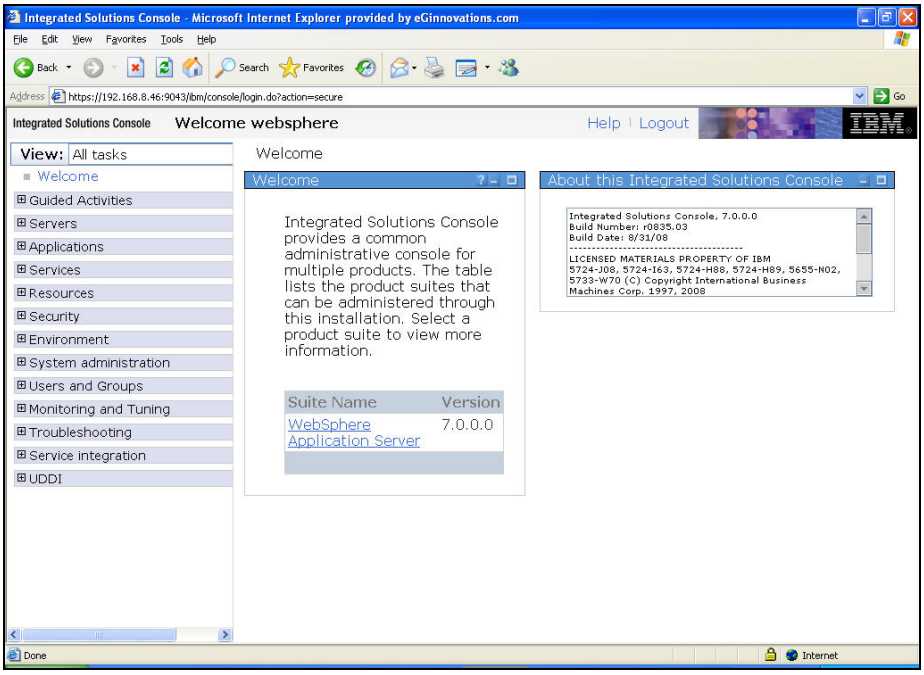

Figure 3.18: The WebSphere Administrative Console

3. Expand the **Applications** node in the tree-structure in the left pane of Figure 3.19, and then click

on the **New Application** sub-node within. Then, from the list of links displayed in the right panel, pick the **New Enterprise Application** link.

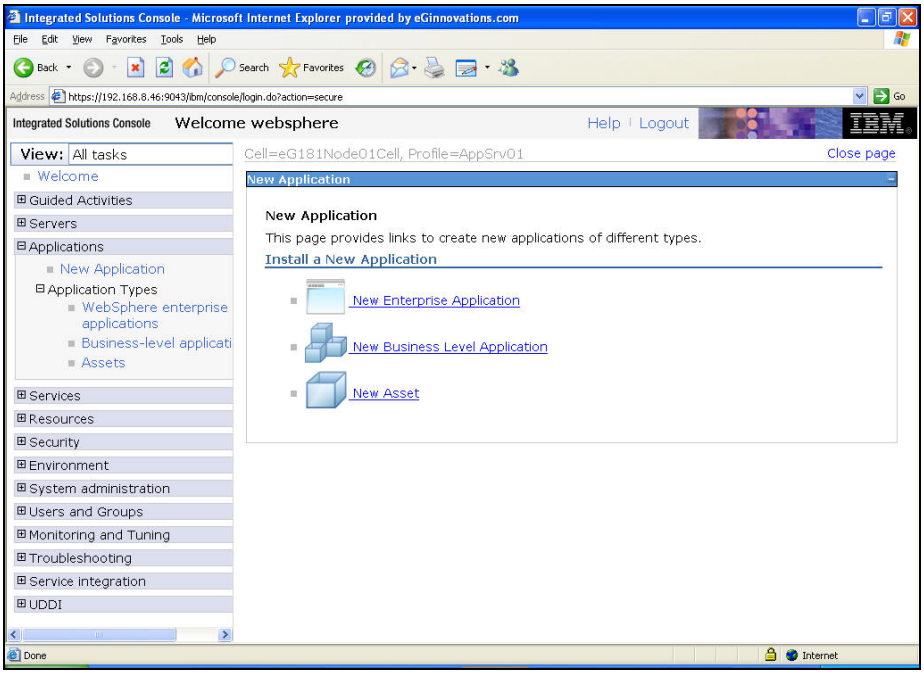

Figure 3.19: Installing a new application

4. Figure 3.20 will then appear. If the **egurkha.ear** file to be deployed is on the local host, then, select the **Local file system** option, and then specify the full path to the **egurkha.ear** application in the **Full path** text box. On the other hand, if the **egurkha.ear** file is available on a remote host in the environment, then, pick the **Remote file system** option and specify the full path to the remote **egurkha.ear** file in the **Full path** text box below. Typically, the 'ear' file will be available in the **/opt/egurkha/lib** directory (on Unix; on Windows, it will be the <EG_INSTALL_DIR>\lib directory). Then, click the **Next** button in Figure 3.20.

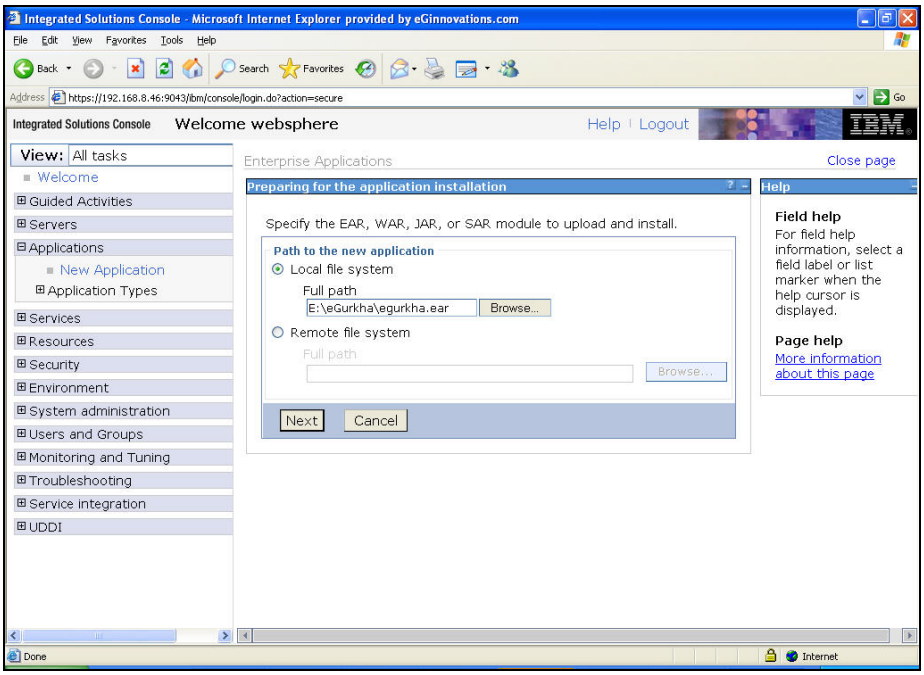

Figure 3.20: Specifying the full path to the 'egurkha.ear' file

5. When Figure 3.21 appears, select the **Fast Path** option, and click the **Next** button.

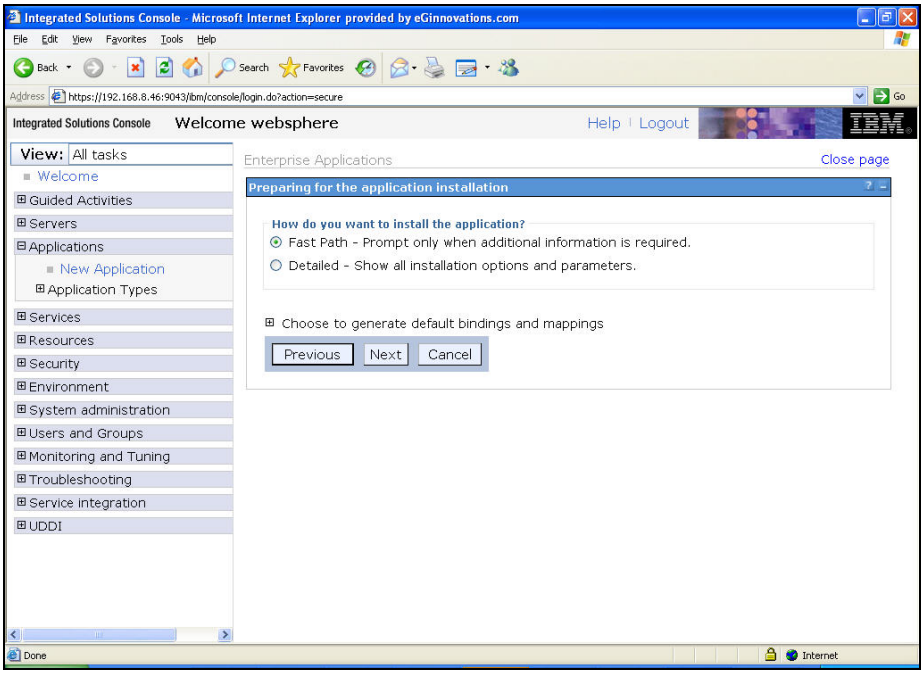

Figure 3.21: Indicating how to install the egurkha application

6. Using Figure 3.22 that appears next, view the **Installation Options**, and select the options that

you want to enable for your application deployment. Since the default installation settings apply to the **egurkha.ear** deployment, you need not enable/disable any additional options in this page. Therefore, simply scroll down the page and click the **Next** button therein.

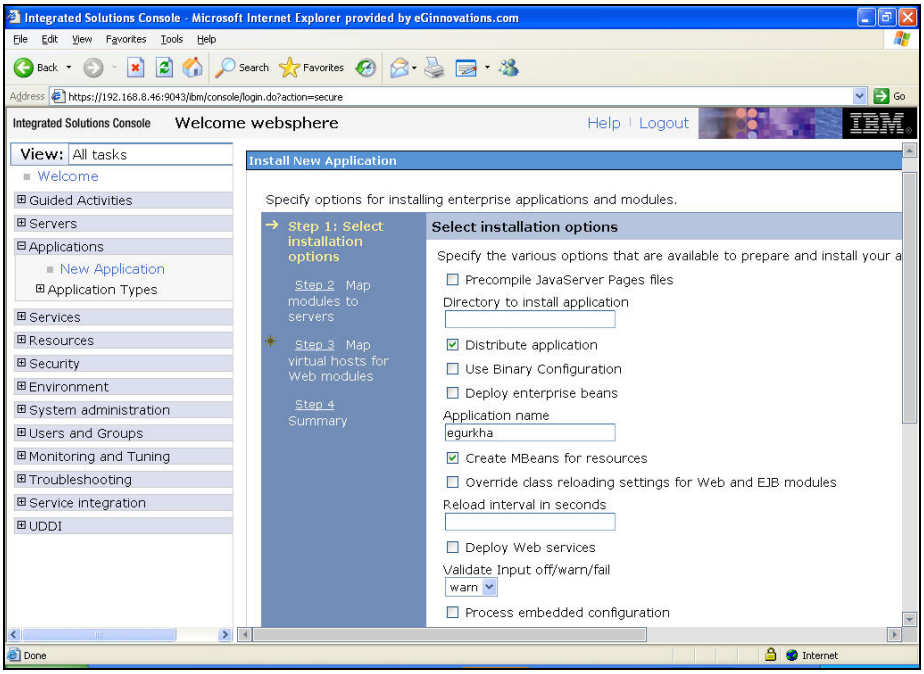

Figure 3.22: Viewing the installation options

7. When Figure 3.23 appears, map the **egurkha** module to the displayed **Server**, by selecting the check box corresponding to the **egurkha** module and clicking the **Next** button.

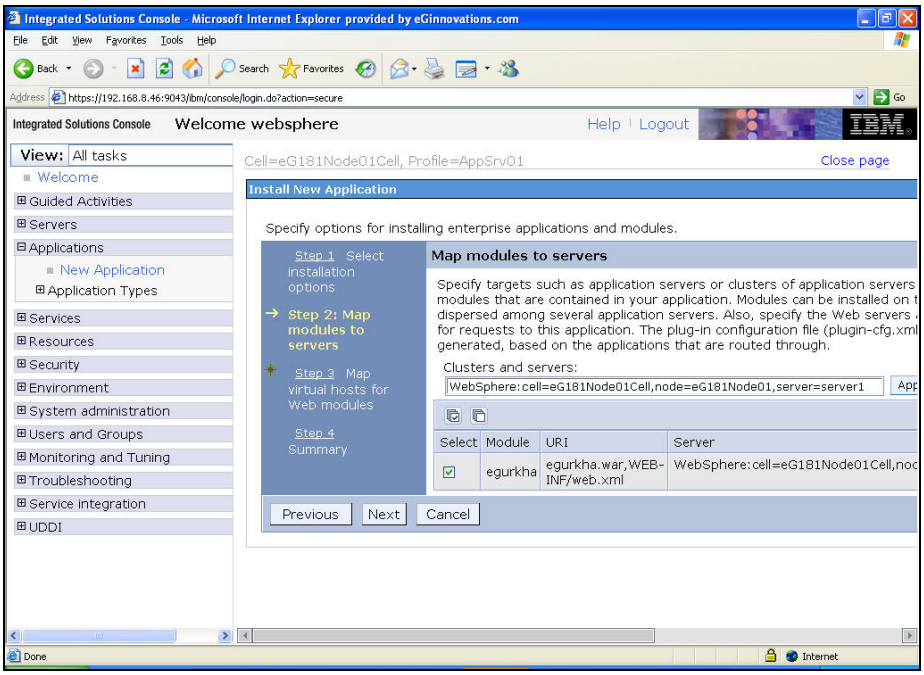

Figure 3.23: Mapping the egurkha module with a server

8. Then, to map the **egurkha** web module to the default selection in the **Virtual Host** drop-down, select the check box corresponding to the **egurkha** module in Figure 3.25, and click the **Next** button. This will ensure that the **egurkha** web module is installed on the default virtual host.

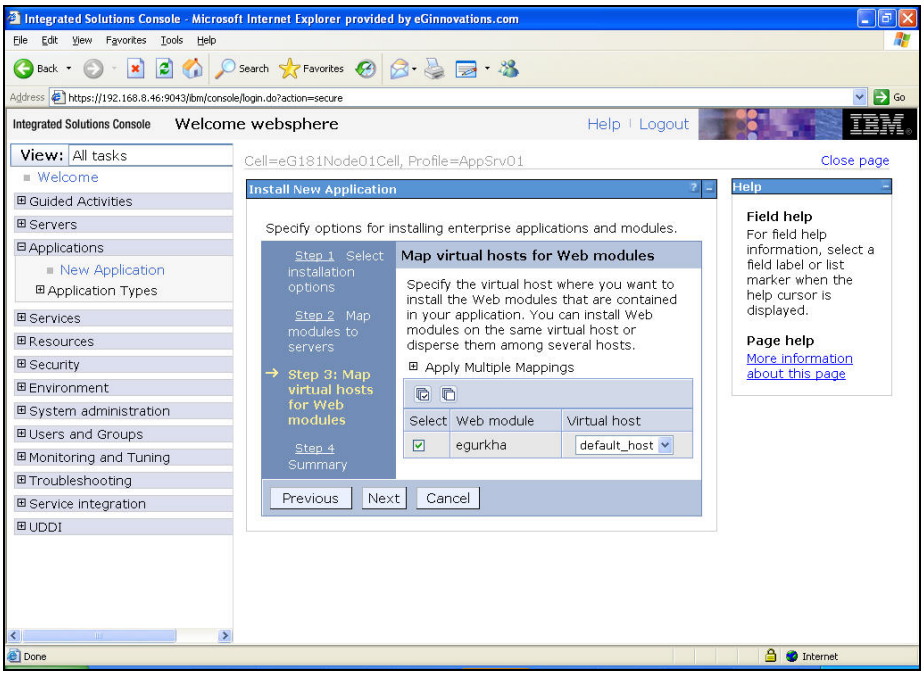

Figure 3.24: Mapping the egurkha web module with a virtual host

9. Next, a summary of your specifications for **egurkha.ear** deployment will appear (see Figure 3.26). Scroll down the summary to click the **Finish** button.

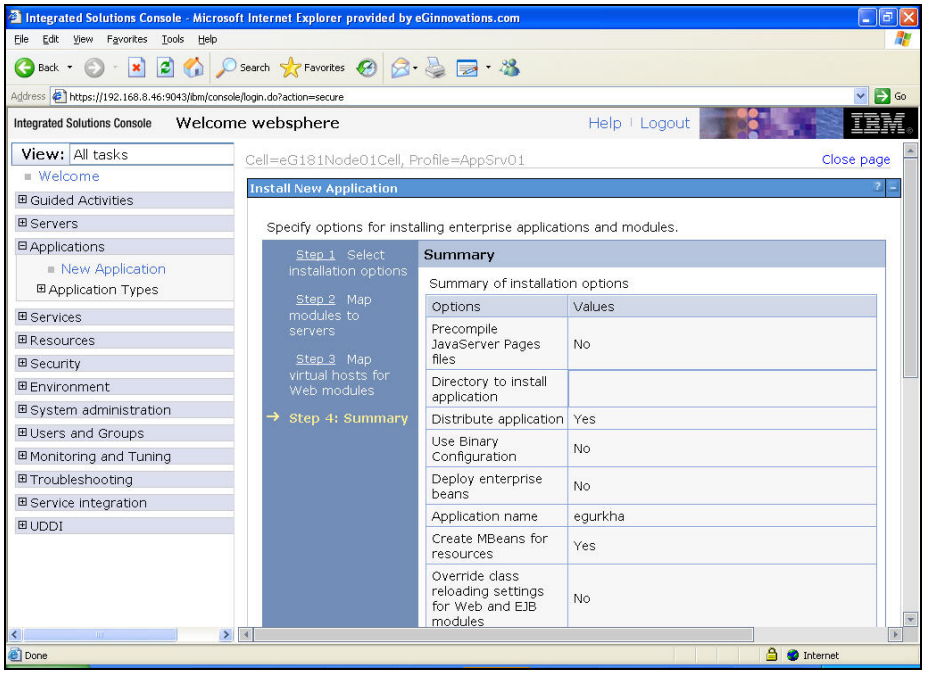

Figure 3.25: Viewing a summary of your installation specifications

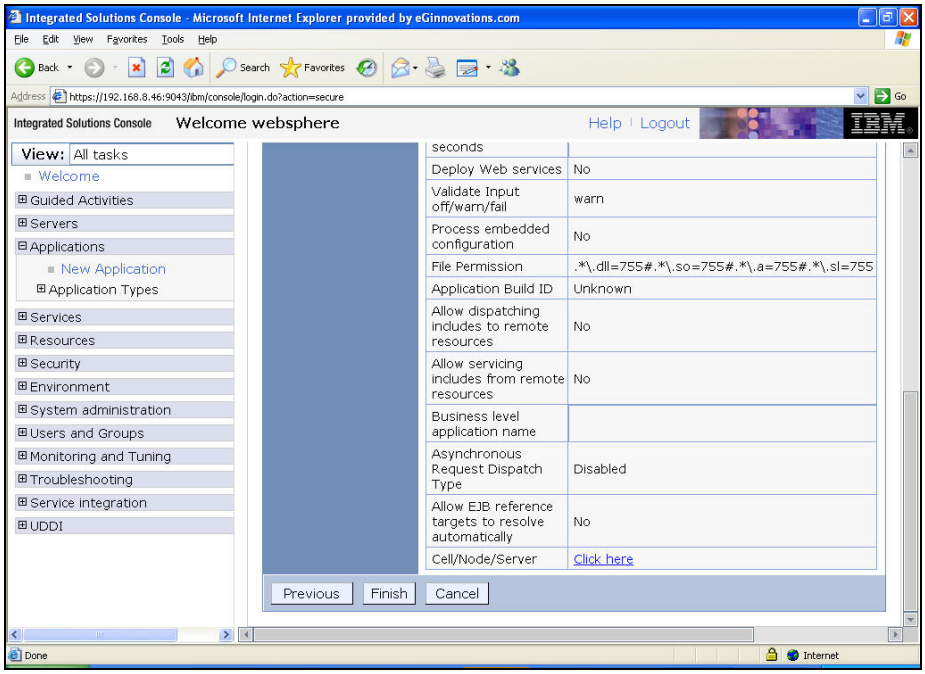

Figure 3.26: Scrolling down the summary and clicking the Next button

10. This will trigger the deployment of the **egurkha** application. Figure 3.27 indicates the status of the installation.

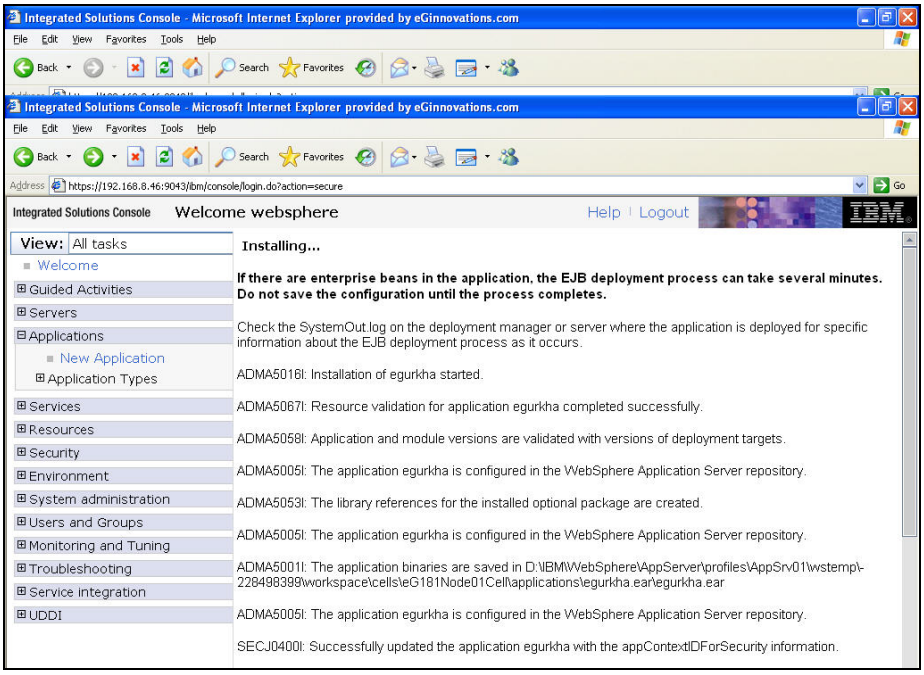

Figure 3.27: Status of the egurkha installation

11. Scroll down Figure 3.27 to determine whether the **egurkha** application was successfully deployed or not. If application deployment is successful, a message to that effect will appear here (see Figure 3.28).

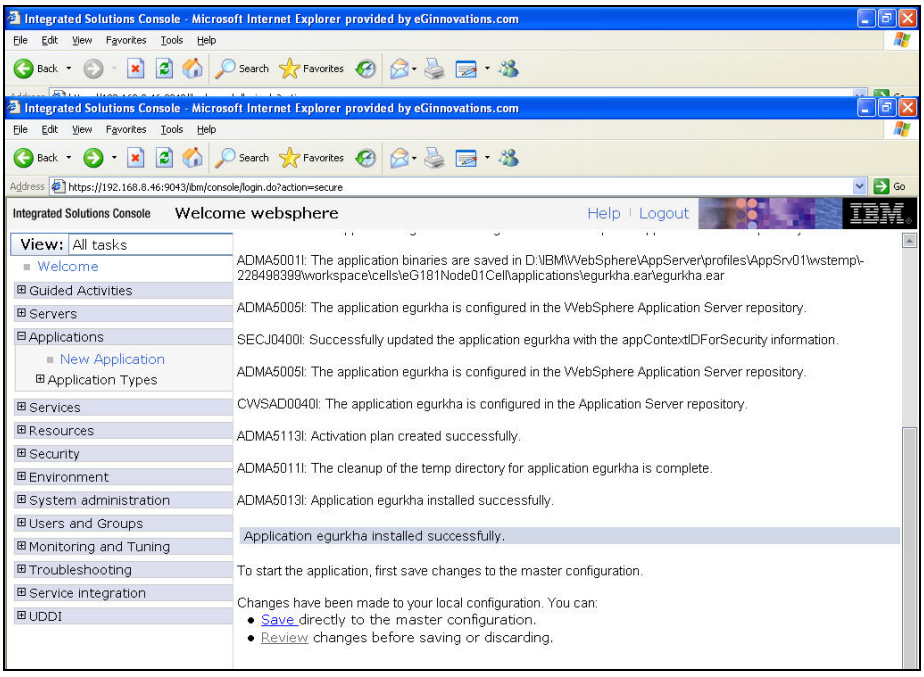

Figure 3.28: A message indicating that the egurkha application has been successfully installed

12. Then, proceed to save the **egurkha** application. The first step towards this is to save the changes to the master configuration. Before saving the master configuration, you may want to review the changes effected on it. For this, click the **Review** link in Figure 3.28.
13. Figure 3.29 will then appear displaying the **Total** number of **changed documents**. To view the list of documents that changed, expand the **Total changed documents** node (see Figure 3.30).

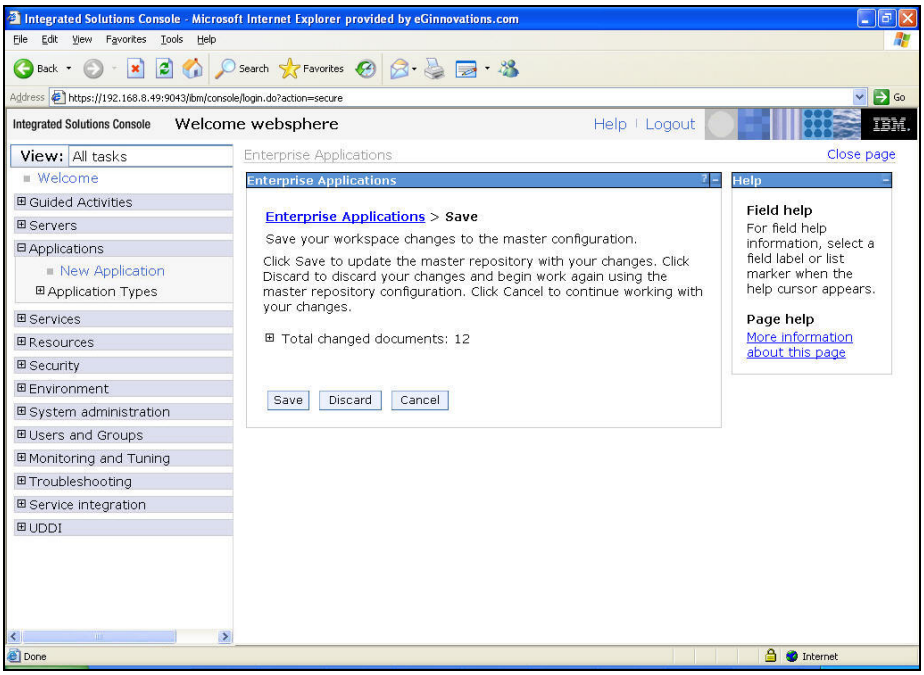

Figure 3.29: Viewing the total number of changed documents

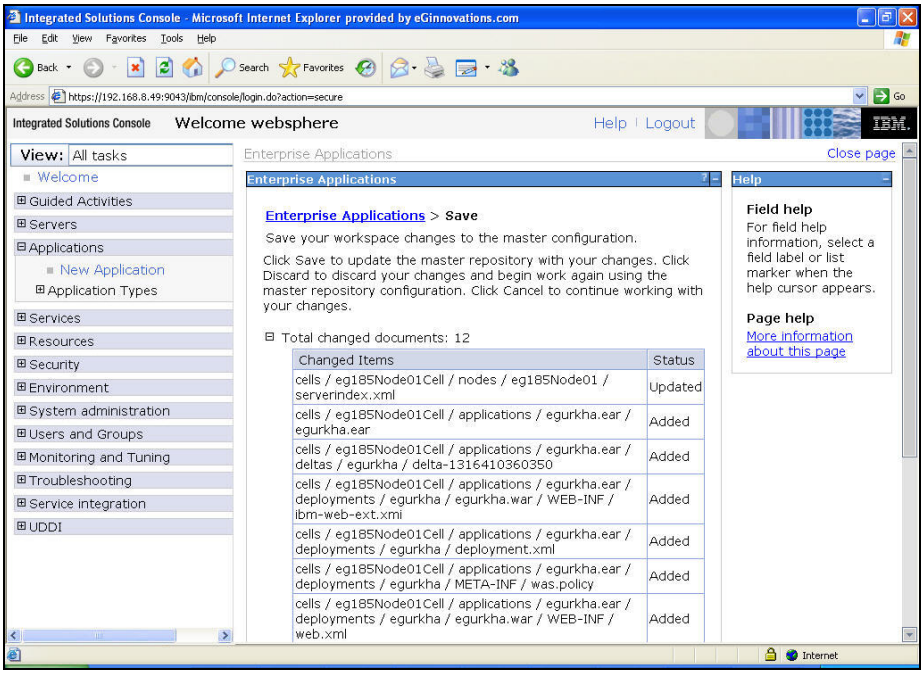

Figure 3.30: Reviewing the changes to the documents

14. Scroll down Figure 3.31 to review the complete list of changes. Finally, click the **Save** button in Figure 3.31 to save the changes.

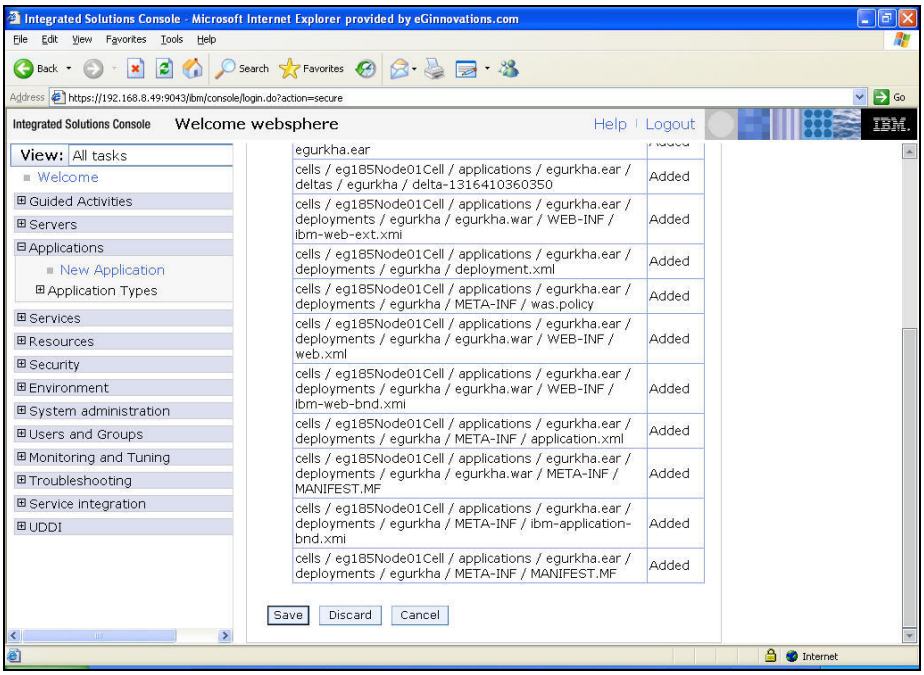

Figure 3.31: Saving the changes

15. Now, proceed to start the **egurkha** application. For that, follow the **Applications -> Application Types -> WebSphere enterprise applications** node sequence in the tree-structure in the left panel of Figure 3.32. The list of installed applications will then appear in the right panel. Select the check box corresponding to the **egurkha** application and click the **Start** button therein to start it.

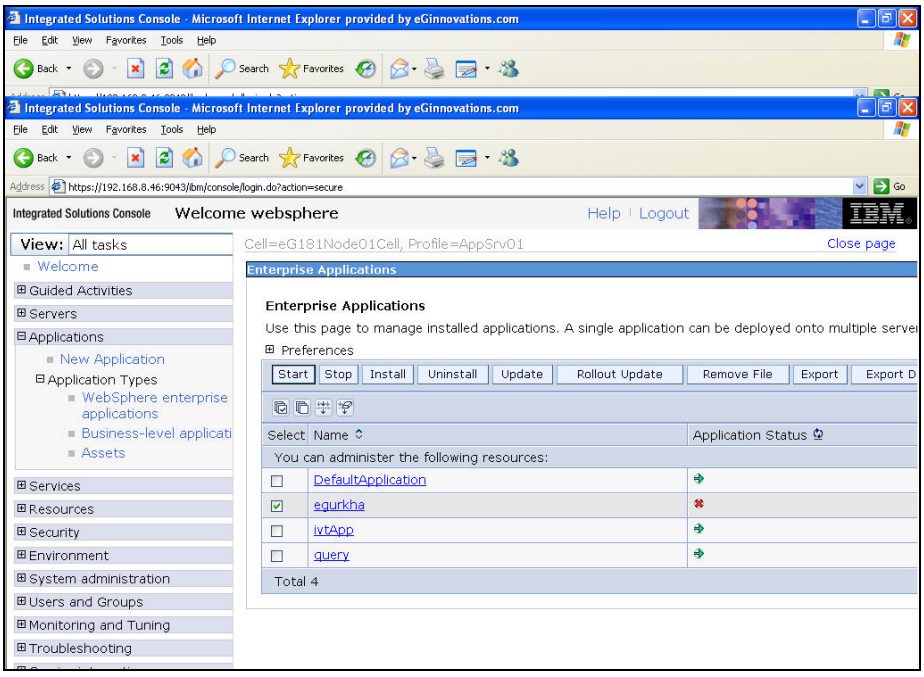

Figure 3.32: Selecting the Save to Master Configuration link

16. Once the application is started, a message to that effect will appear as depicted by Figure 3.33 below.

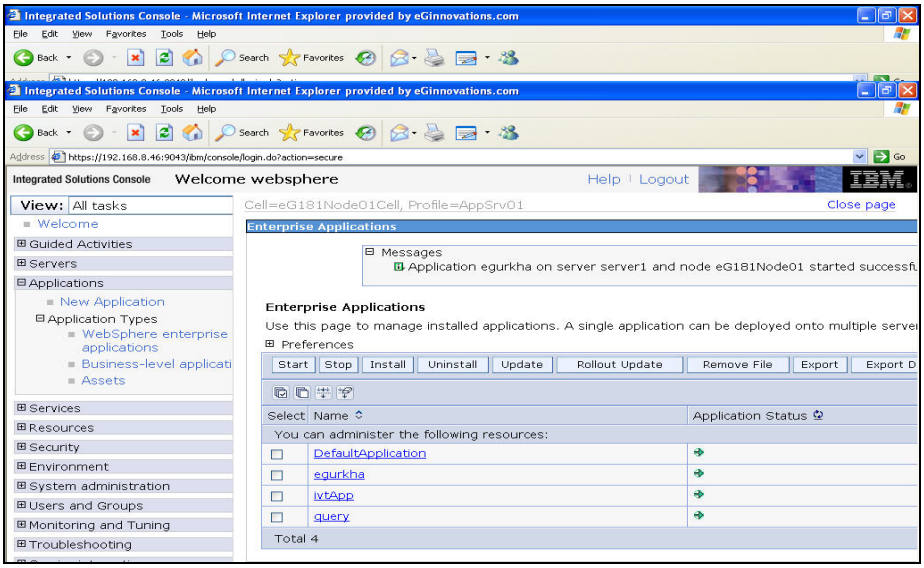

Figure 3.33: A message indicating that the egurkha application has started successfully

17. Finally, log out of the admin console.

3.3 Configuring a WebSphere Application Server 8.x to Work with the eG Agent

To monitor a WebSphere Application server 6.x (and above), a specific WebSphere monitoring eG component has to be installed on it. If the WebSphere server to be monitored uses a SOAP connector port for its internal communications, then the **egurkha6.ear** component will have to be installed on the server. On the other hand, if the target WebSphere server uses an RMI port for communicating internally, then the **egurkha6rmi.ear** component should be installed on the server. Before installing either of these components on a WebSphere server, do the following:

- Copy the **egurkha6.ear** (or **egurkha6rmi.ear**) file from the **<EG_INSTALL_DIR>\lib** directoy (on Windows; on Unix, this will be **/opt/egurkha/lib**) to any temporary location on the eG agent host.
- Rename the file in the temporary folder as **egurkha.ear**.
- Copy the **egurkha.ear** file back to the <EG_INSTALL_DIR>\lib directoy (on Windows; on Unix, this will be **/opt/egurkha/lib**).

Then, follow the steps given below to perform the installation.

1. Login to the WebSphere Administrative Console by first accessing the URL: http://<IP/hostname of the WebSphere server>:<WebSpherePort>/, and then providing a valid **User ID** and **Password**.

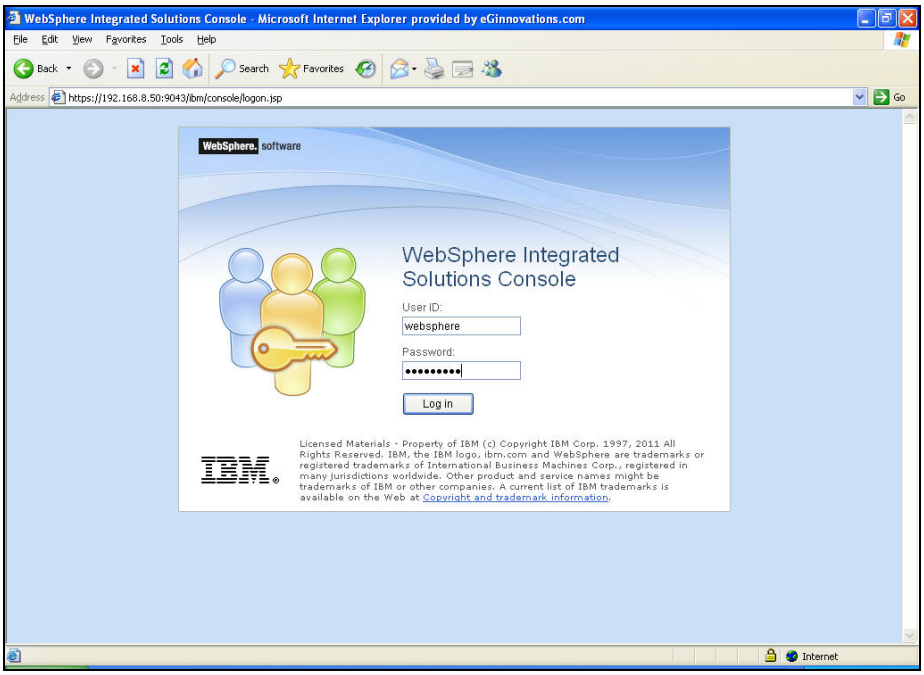

Figure 3.34: Logging into the WebSphere administrative console

2. The **WebSphere Administrative Console** will then appear as depicted by Figure 3.35.

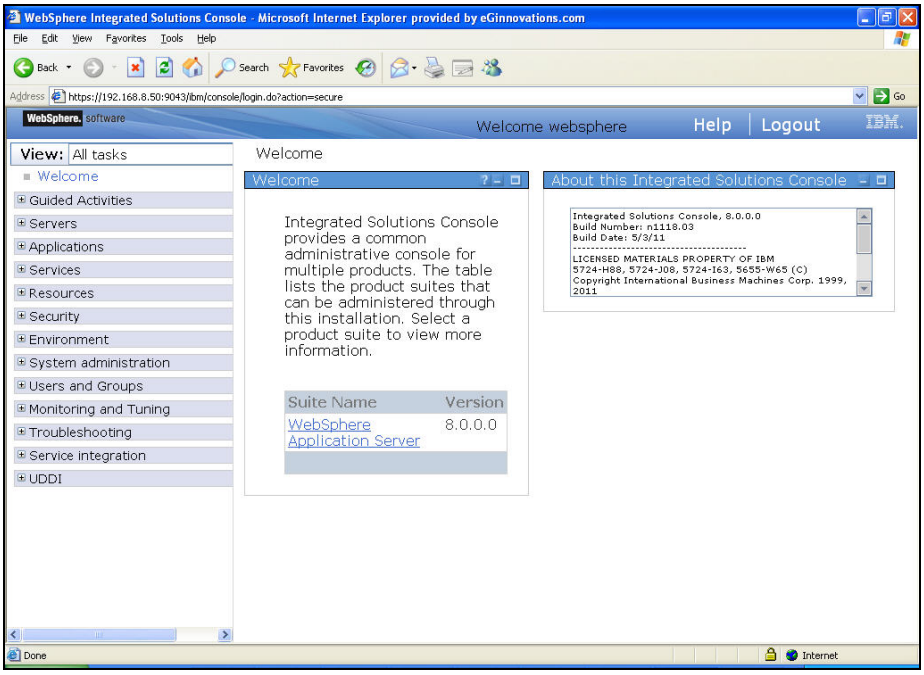

Figure 3.35: The WebSphere Administrative Console

3. Expand the **Applications** node in the tree-structure in the left pane of Figure 3.36, and then click

on the **New Application** sub-node within. Then, from the list of links displayed in the right panel, pick the **New Enterprise Application** link.

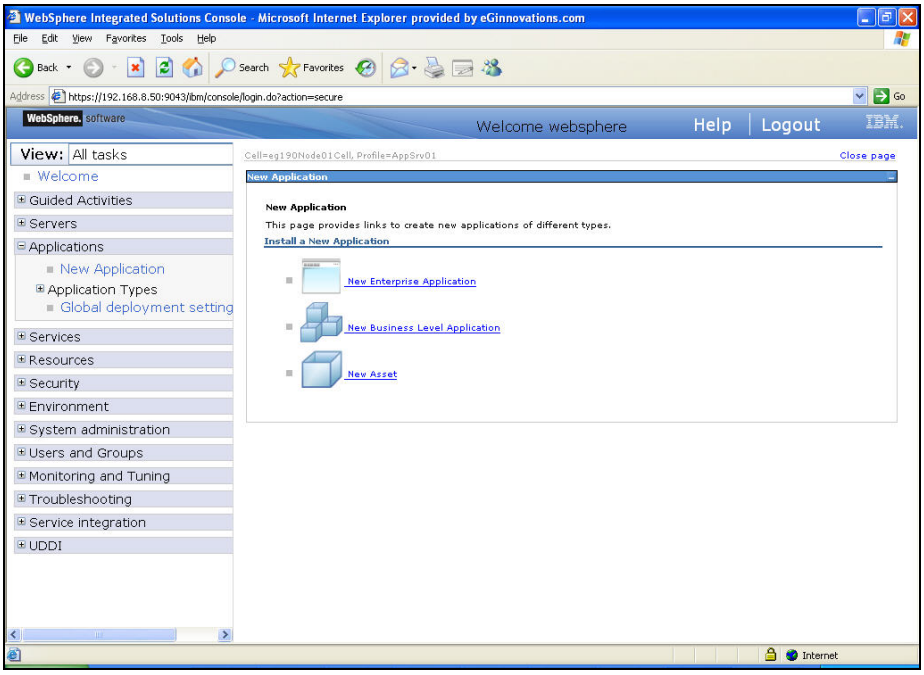

Figure 3.36: Installing a new application

4. Figure 3.37 will then appear. If the **egurkha.ear** file to be deployed is on the local host, then, select the **Local file system** option, and then specify the full path to the **egurkha.ear** application in the **Full path** text box. On the other hand, if the **egurkha.ear** file is available on a remote host in the environment, then, pick the **Remote file system** option and specify the full path to the remote **egurkha.ear** file in the **Full path** text box below. Typically, the 'ear' file will be available in the **/opt/egurkha/lib** directory (on Unix; on Windows, it will be the **<EG_INSTALL_DIR>\lib** directory). Then, click the **Next** button in Figure 3.37.

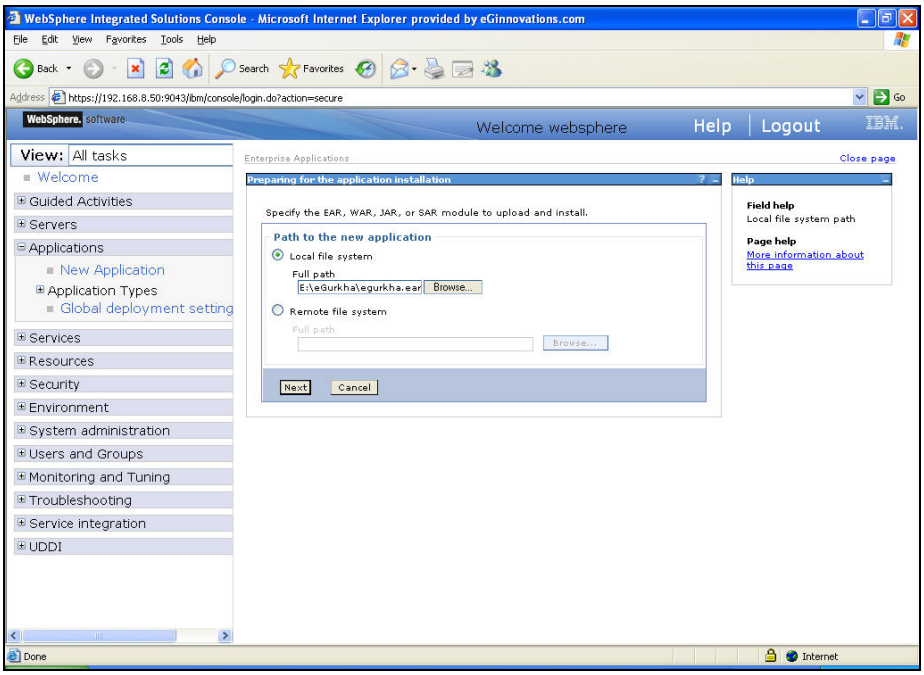

Figure 3.37: Specifying the full path to the 'egurkha.ear' file

5. When Figure 3.38 appears, select the **Fast Path** option, and click the **Next** button.

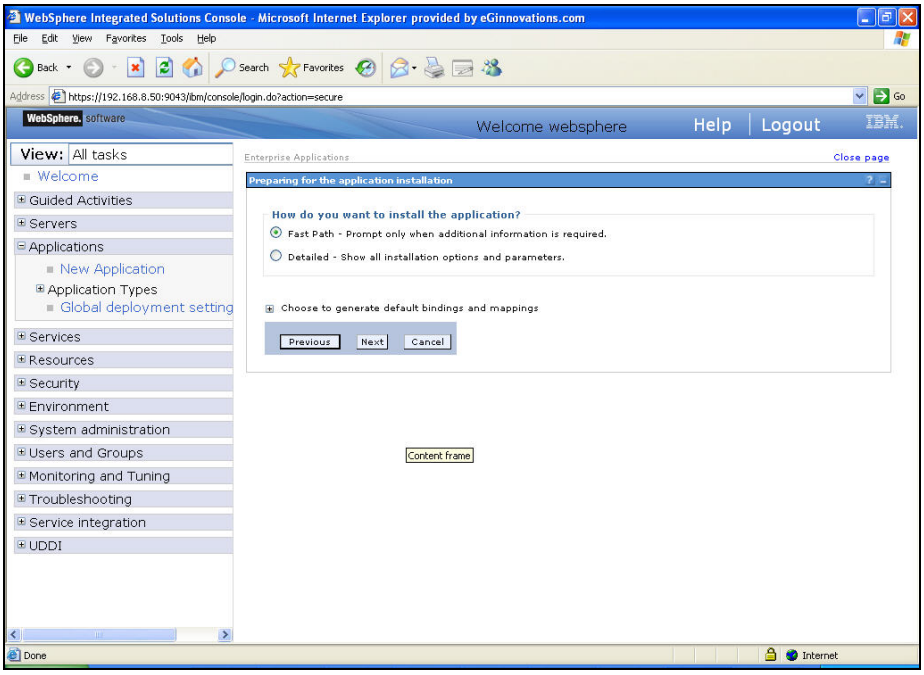

Figure 3.38: Indicating how to install the egurkha application

6. Using Figure 3.39 that appears next, view the **Installation Options**, and select the options that

you want to enable for your application deployment. Since the default installation settings apply to the **egurkha.ear** deployment, you need not enable/disable any additional options in this page. Therefore, simply scroll down the page and click the **Next** button therein.

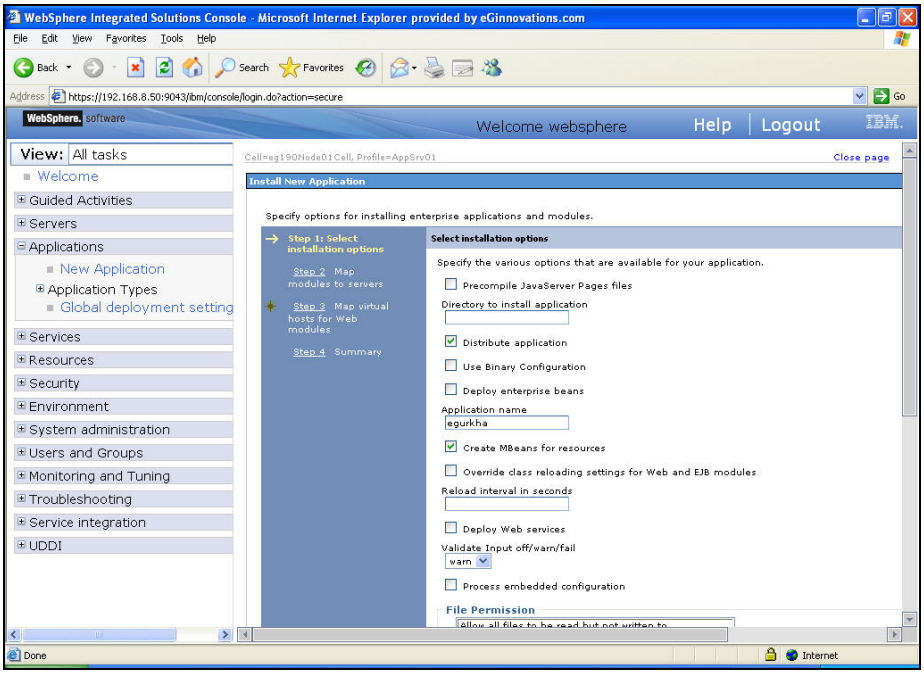

Figure 3.39: Viewing the installation options

7. When Figure 3.40 appears, map the **egurkha** module to the displayed **Server**, by selecting the check box corresponding to the **egurkha** module and clicking the **Next** button.

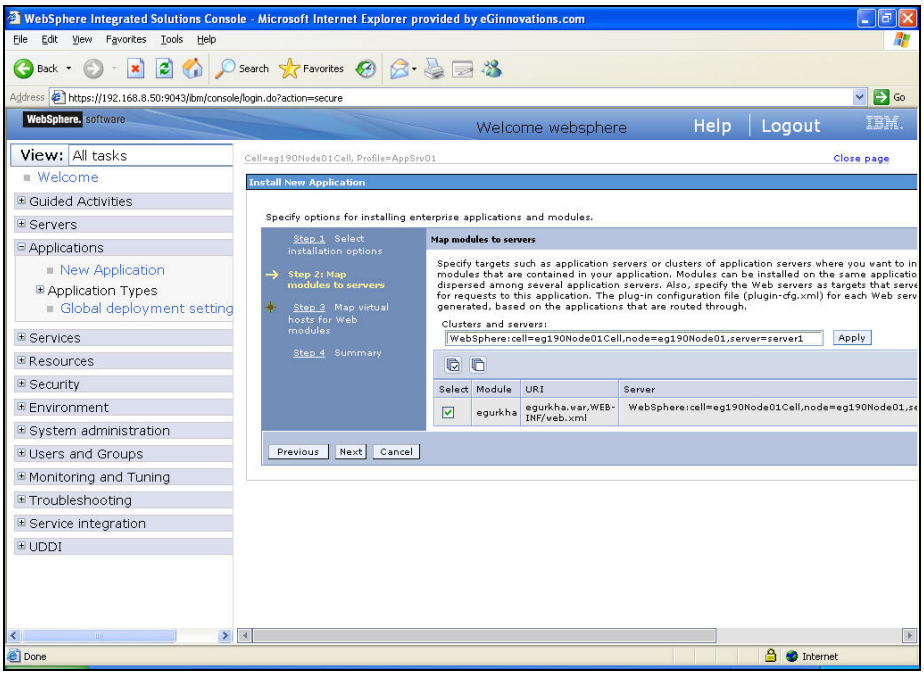

Figure 3.40: Mapping the egurkha module with a server

8. Then, to map the **egurkha** web module to the default selection in the **Virtual Host** drop-down, select the check box corresponding to the **egurkha** module in Figure 3.41, and click the **Next** button. This will ensure that the **egurkha** web module is installed on the default virtual host.

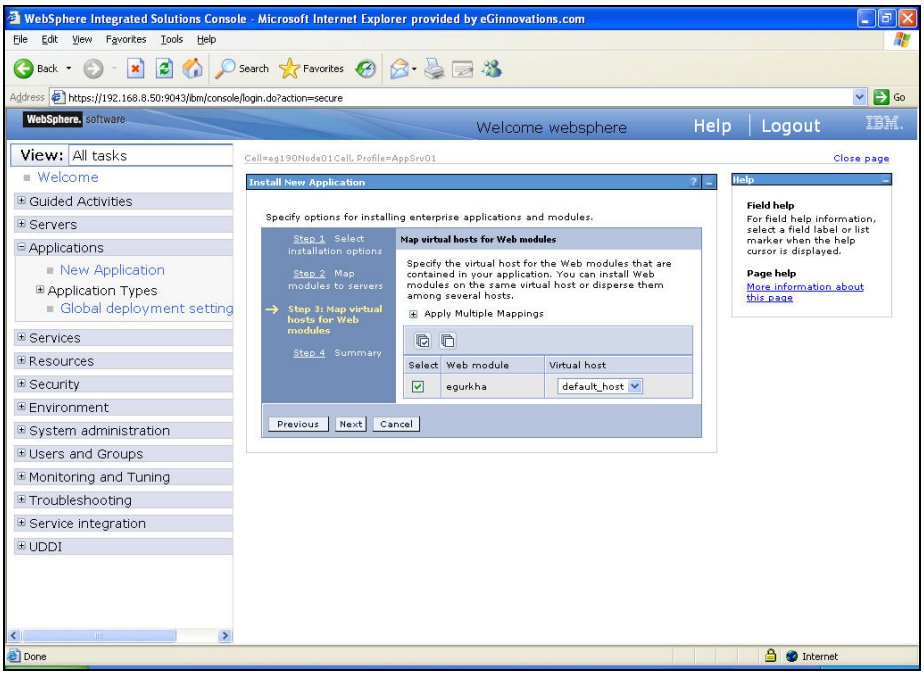

Figure 3.41: Mapping the egurkha web module with a virtual host

9. Next, a summary of your specifications for **egurkha.ear** deployment will appear (see Figure 3.42). Scroll down the summary to click the **Finish** button (see Figure 3.43).

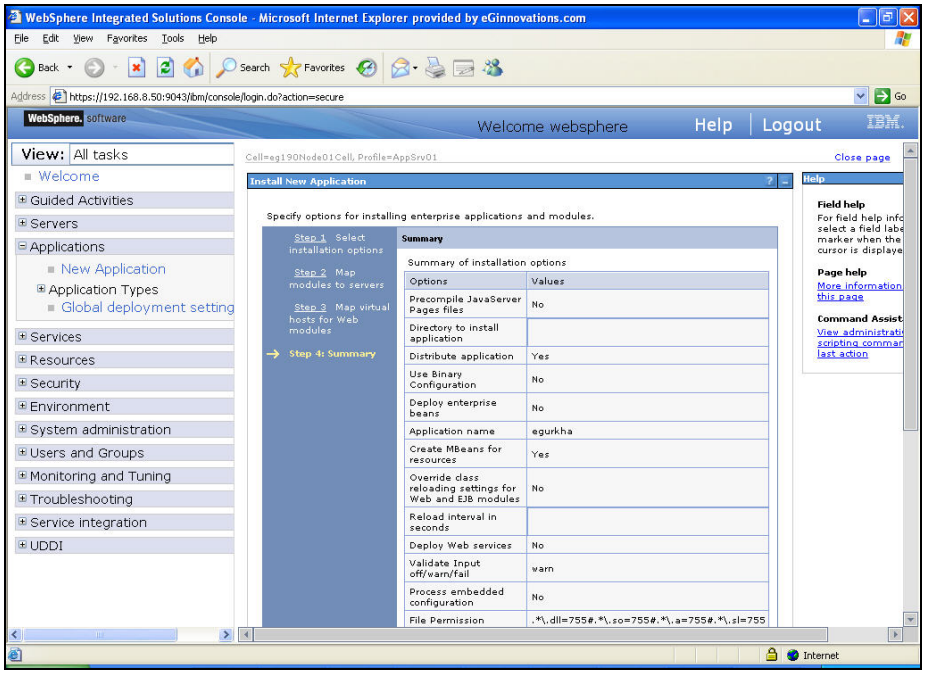

Figure 3.42: Viewing a summary of your installation specifications

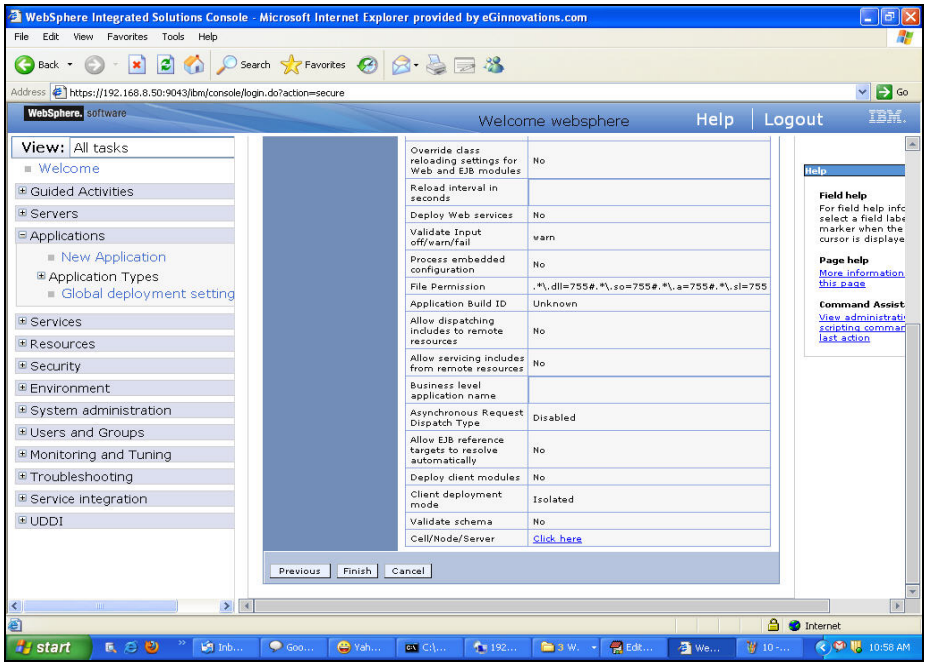

Figure 3.43: Scrolling down the summary and clicking the Next button

10. This will trigger the deployment of the **egurkha** application. Figure 3.44 then appears indicating the progress of the installation. Scroll down Figure 3.44 to determine whether the **egurkha** application was successfully deployed or not. If application deployment is successful, a message to that effect will appear here.

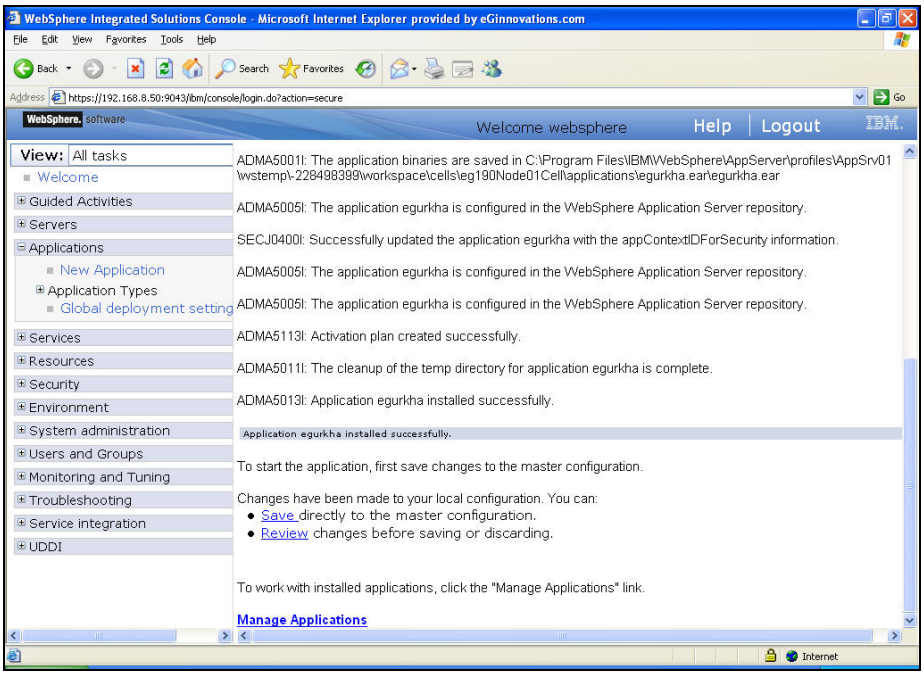

Figure 3.44: Status of the egurkha installation

11. Then, proceed to save the **egurkha** application. The first step towards this is to save the changes to the master configuration. Before saving the master configuration, you may want to review the changes effected on it. For this, click the **Review** link in Figure 3.44.

12. Figure 3.45 will then appear displaying the **Total** number of **changed documents**. To view the list of documents that changed, expand the **Total changed documents** node. To proceed with saving the application, click the **Save** button in Figure 3.45.

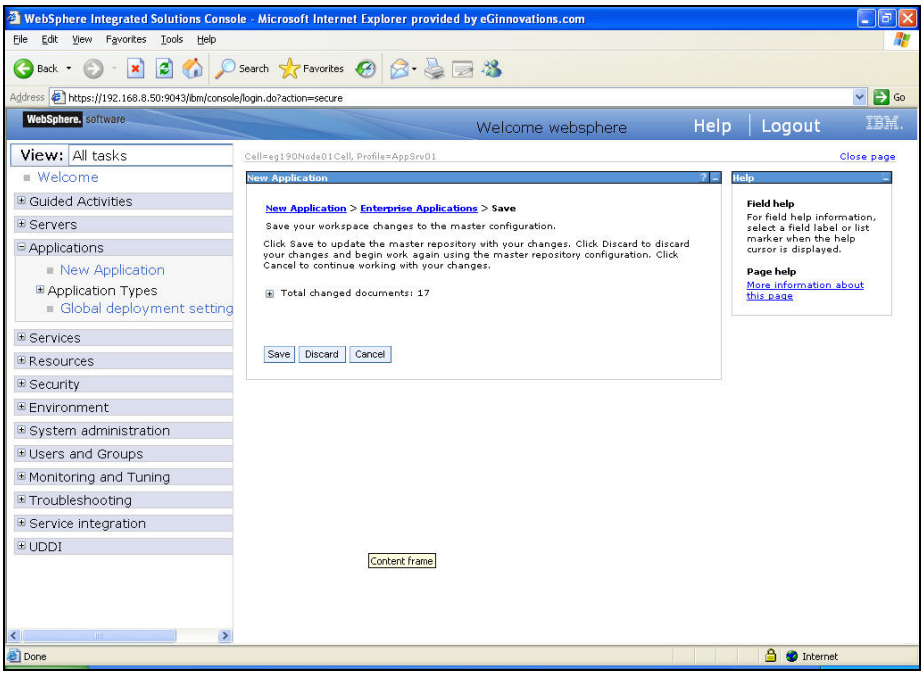

Figure 3.45: Viewing the total number of changed documents

13. Now, proceed to start the **egurkha** application. For that, follow the **Applications -> Application Types -> WebSphere enterprise applications** node sequence in the tree-structure in the left panel of Figure 3.45. The list of installed applications will then appear in the right panel. Select the check box corresponding to the **egurkha** application and click the **Start** button therein to start it (see Figure 3.46).

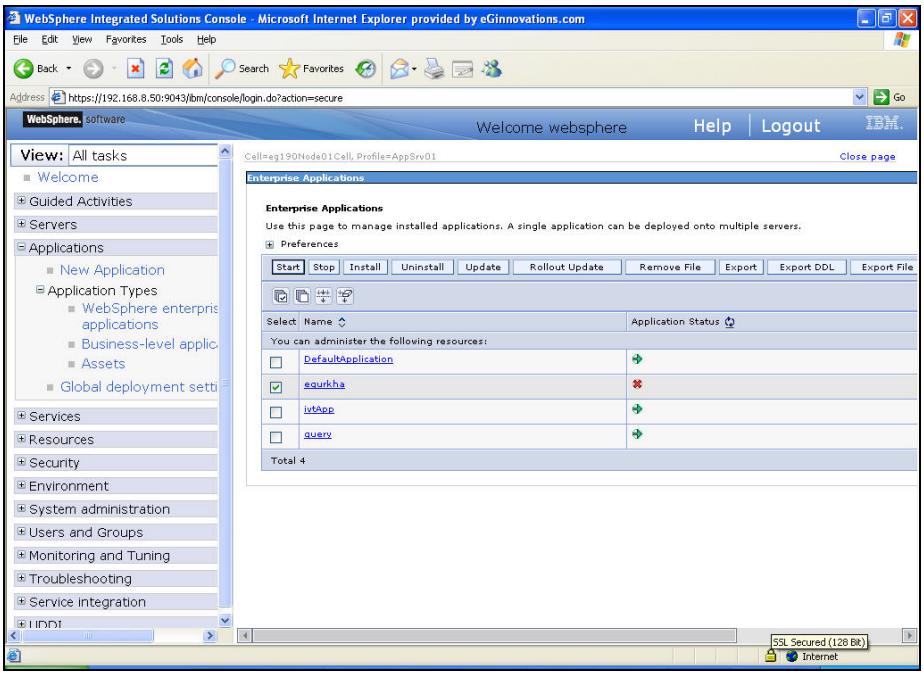

Figure 3.46: Starting the egurkha application

14. Once the application is started, a message to that effect will appear as depicted by Figure 3.47 below.

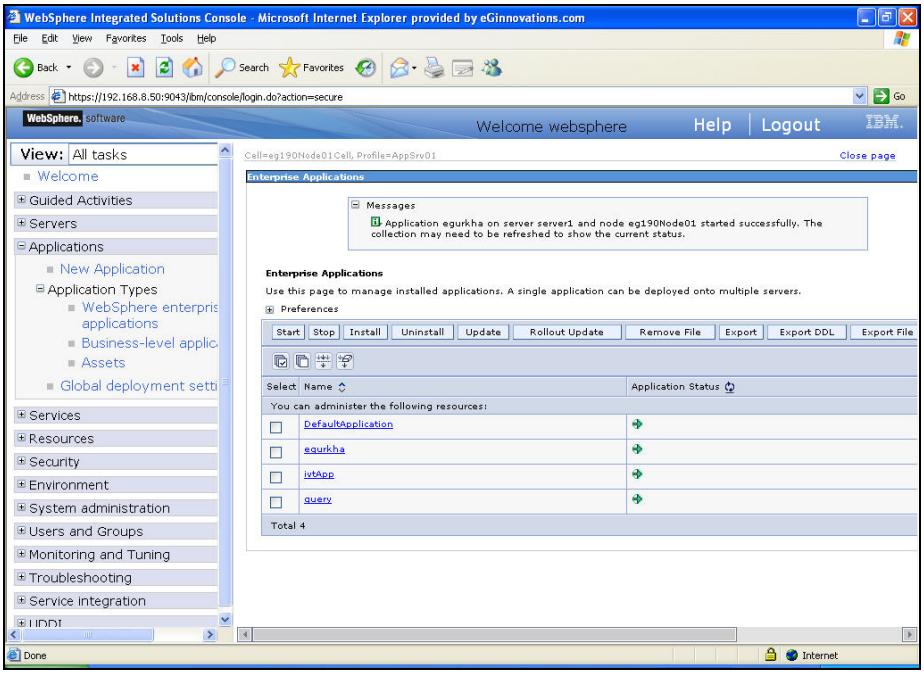

Figure 3.47: A message indicating that the egurkha application has started successfully

15. Finally, log out of the admin console.

3.4 Enabling Performance Monitoring

To enable performance monitoring for the WebSphere server 6.0 (or above), do the following:

1. Open WebSphere's admin console as explained in steps 1-3 of Section **3.1**.
2. In the screen that appears (see Figure 3.48), expand the **Monitoring and Tuning** node in the left pane, and click on the **Performance Monitoring Infrastructure (PMI)** option within.

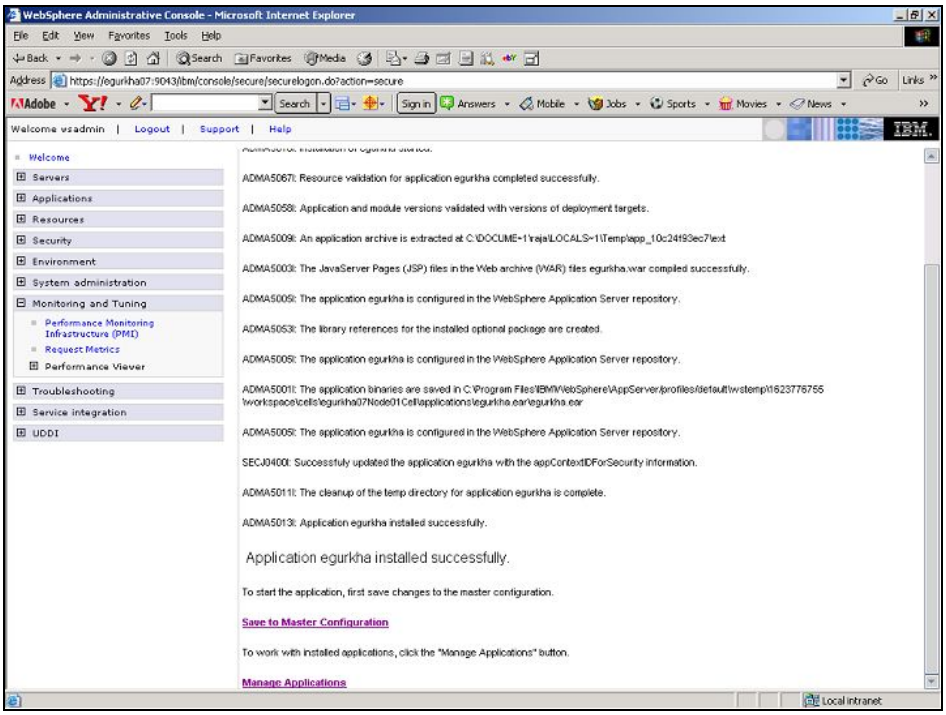

Figure 3.48: Selecting the 'Performance Monitoring Infrastructure (PMI)' option

3. As you can see, the right pane of Figure 3.49 displays the list of existing application servers. Now, click on the server that has been configured for monitoring by eG as indicated by Figure 3.49.

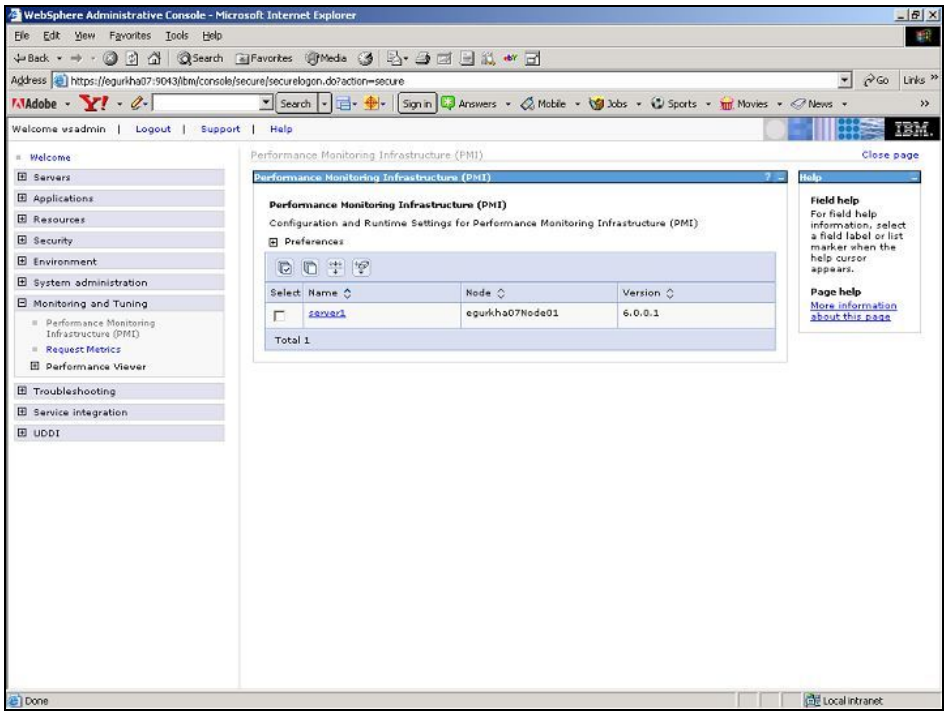

Figure 3.49: Clicking on the server that has been configured for monitoring by eG

4. In Figure 3.50 that appears, select the **Enable Performance Monitoring Infrastructure (PMI) option**, and then click the **Apply** and **OK** buttons to register the changes.

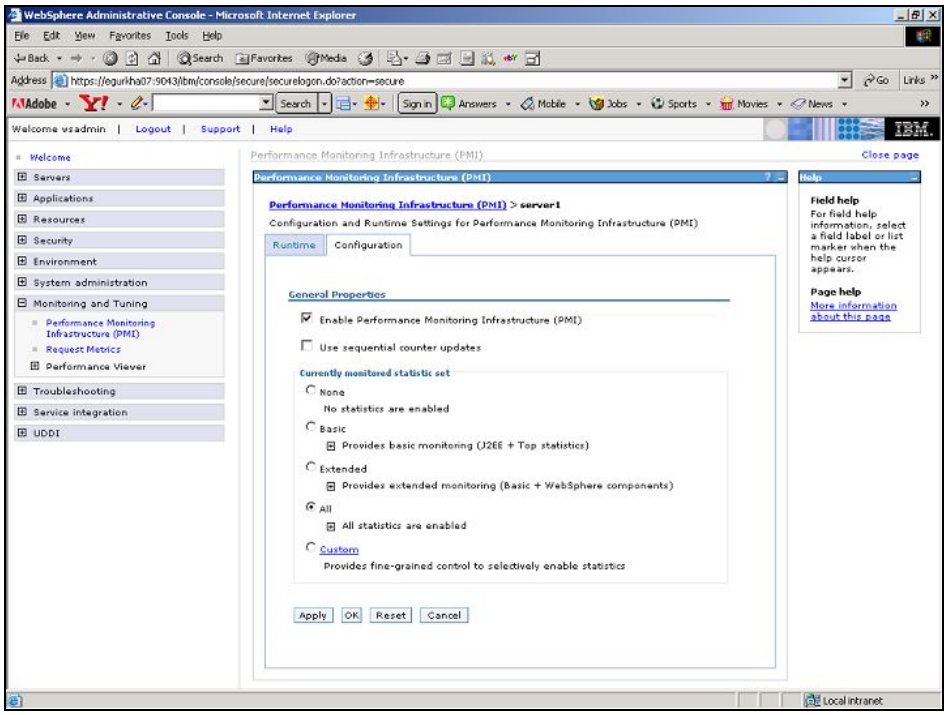

Figure 3.50: Enabling Performance Monitoring

5. Finally, restart the WebSphere server. For that, first stop the server using the command **./stopServer.sh <servername>**, and then, start it using the command **./startServer.sh <servername>**. Both these commands are available in the **<WAS_HOME>/bin** directory, where **<WAS_HOME>** is the install directory of the WebSphere server.

3.5 Enabling the WASJVM Test to Collect GC and Thread-related Statistics

You should enable the Java Virtual Machine Tool Interface (JVMTI) to collect data about garbage collection and thread states from the JVM that runs the application server. To enable JVMTI, follow the steps below:

1. Open the administrative console.
2. Click **Servers > Server Types > WebSphere application servers** in the administrative console navigation tree.
3. Click the application server for the JVM profiler that needs enabling.
4. In the Server Infrastructure section, click Java and process management > Process definition , click Control , and then click Java virtual machine.
5. Type -agentlib:pmiJvmtiProfiler in the **Generic JVM arguments** field. In a 64-bit environment,

type -agentlib:pmiJvmtiProfiler64.

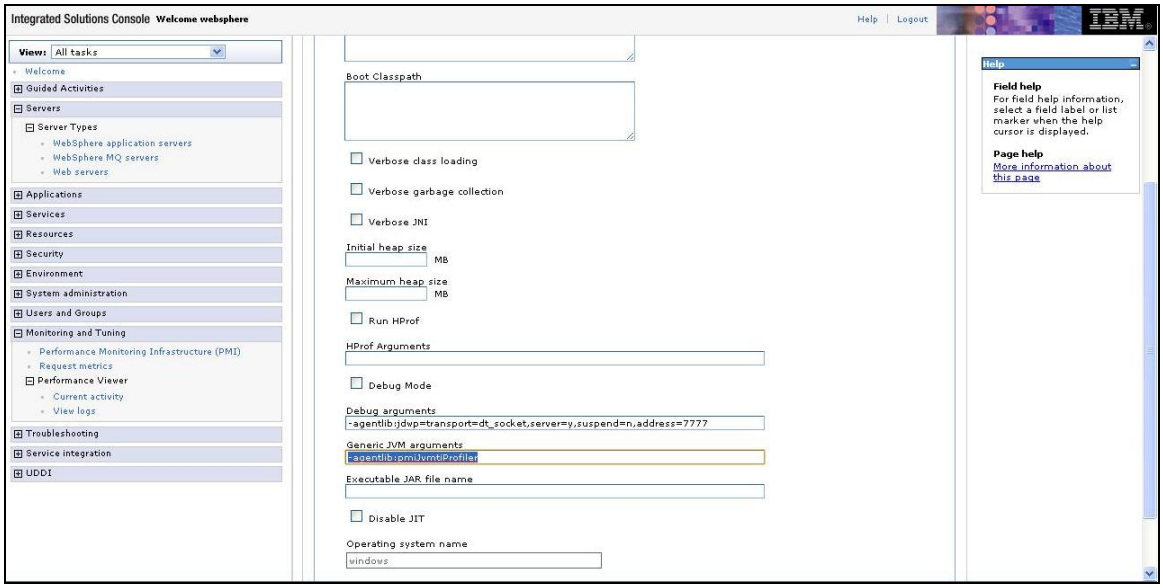

Figure 3.51: Enabling JVMTI

6. Click **Apply** or **OK**.
7. Click **Save**.
8. Click **Servers > Server Types > WebSphere application servers** in the administrative console navigation tree.
9. Click the application server for the JVM profiler that needs enabling.
10. Click the **Configuration** tab. When in the **Configuration** tab, settings apply when the server is restarted. When in the **Runtime** Tab, settings apply immediately. Performance Monitoring Infrastructure (PMI) can be enabled only in the **Configuration** tab.
11. Click Performance Monitoring Infrastructure under Performance.
12. Select the Enable Performance Monitoring Infrastructure (PMI) check box.
13. Click **Custom** and select **JVM Runtime** on the left-side tree.
14. Click a JVM profiler group under **JVM Runtime** and enable or disable the statistic on the right-side table. Go back to the main PMI configuration page, by clicking PMI.
15. Click **Apply** or **OK**.
16. Click **Save**.
17. Start the application server, or restart the application server if it is currently running.

3.6 Configuring the eG Agent to Collect JVM-related Metrics from the WebSphere Server 6.1 and 7.0

By default, no tests are mapped to the **JVM** layer of the *WebSphere* monitoring model. If need be, you can map certain key tests with this layer, and configure these tests to report critical statistics related to the WebSphere server's JVM. These statistics typically reveal the following:

- The count of classes loaded/unloaded (Java Classes test)
- JVM thread usage (**JVM Threads** test)
- CPU and memory usage of the JVM (**JVM Cpu Usage** test and **JVM Memory Usage** test)
- The effectiveness of the JVM's garbage collection activity (JVM Garbage Collections test)
- The uptime of the JVM (**JVM Uptime** test)
- Whether JMX is currently enabled/disabled on the target WebLogic server (**JMX Connection to JVM** test)
- The count and status of file descriptors (**JVM File Descriptors** test)

These key tests are disabled by default for the WebSphere server.

These tests, if enabled, connect to the JRE used by the WebSphere application server to pull out the above-mentioned metrics.

Prior to enabling these tests therefore, you should configure the eG agent to connect to the JRE and collect the required metrics, using either of the following methodologies:

- JMX (Java Management Extensions)
- SNMP (Simple Network Management Protocol)

Since both JMX and SNMP support are available for JRE 1.5 and above only, these tests **should be enabled only if the WebSphere server being monitored runs JRE 1.5 and above**.

Once the JRE version requirement is fulfilled, you must proceed to enable the JMX support for the JRE or SNMP support for the JRE, before enabling the tests.

If you choose to use JMX for pulling out the desired metrics from the JRE, then the following broad steps need to be followed:

- First, determine whether the JMX requires no authentication at all, or requires authentication (but no security)
- If JMX does not require authentication, follow the steps below:

- Login to the target WebSphere server.
- Edit the *management.properties* file that is used by the JRE of the target WebSphere server, and configure the following in it:
  - The JMX remote port
  - Whether JMX is SSL-enabled or not
  - Whether JMX requires authentication or not
  - To know how to configure these, refer to the *Monitoring Java Applications* document.
- Save the file.

If the JMX requires authentication (but no security), follow the steps below:

1. Login to the target WebSphere server. If the server is executing on a Windows host, then, login to the host as a local/domain administrator.
2. Next, copy the *jmxremote.password.template* file in the **<JAVA_HOME>\jre\lib\management** folder to any other location on the host, rename it as as *jmxremote.password*, and then, copy it back to the **<JAVA_HOME>\jre\lib\management** folder.
3. Next, edit the *jmxremote.password* file and the *jmxremote.access* file to create a user with *read-write* access to the JMX. To know how to create such a user, refer to *Monitoring Java Applications* document.
4. Then, proceed to make the *jmxremote.password* file secure by granting a single user "full access" to that file. To know how to achieve this, refer to the *Monitoring Java Applications* document.
5. Edit the *management.properties* file that is used by the JRE of the target WebSphere server, and configure the following in it:
   - The JMX remote port
   - Whether JMX is SSL-enabled or not
   - Whether JMX requires authentication or not
   - The full path to the *jmxremote.access* file
   - The full path to the *jmxremote.password* file

   To know how to configure these, refer to the *Monitoring Java Applications* document.
6. Then, save the file.

7. Connect to the Admin console of the target WebSphere server using the URL: **http://<WebSphereServerIP>:<WebSpherePort>**
8. The default port number of the WebSphere server is 9062.
9. Provide the credentials for logging into the admin console, if prompted.
10. Once the console opens, follow the node sequence, *Server -> Server Types -> WebSphere application servers*, in the tree-structure in the left pane of the console.

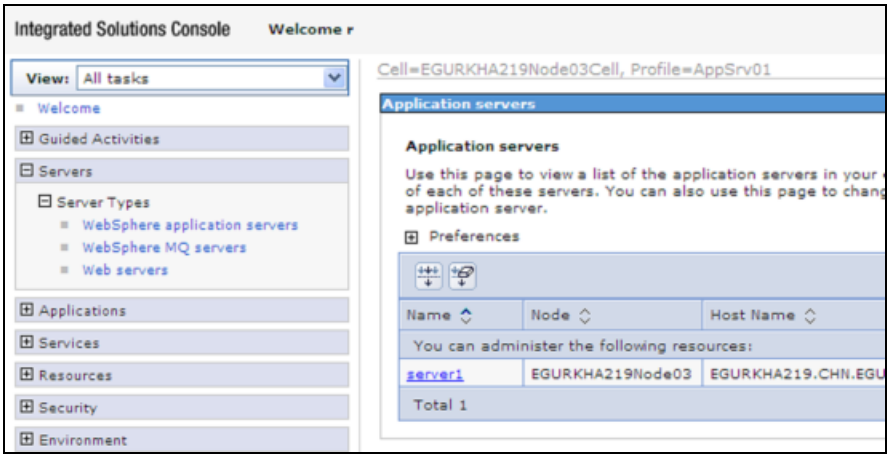

Figure 3.52: Viewing the WebSphere application servers

11. From the list of application servers listed in the right panel, choose the server being monitored by clicking on it.
12. Then, follow the node sequence Server Infrastructure -> Java and Process Management -> Process definition -> Java Virtual Machine, as depicted by Figure 3.53 and Figure 3.54 below:

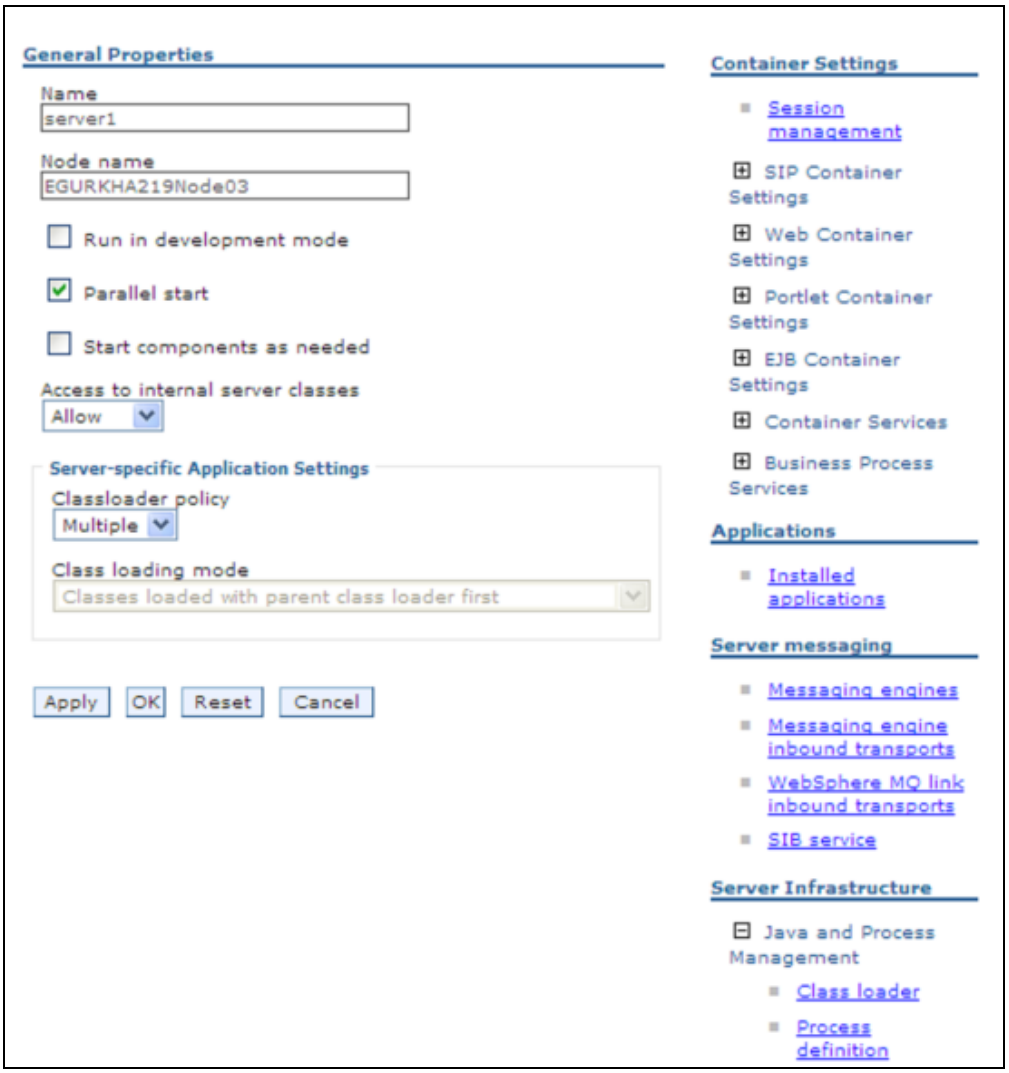

Figure 3.53: Navigating to the Java Virtual Machine properties of the target WebSphere server

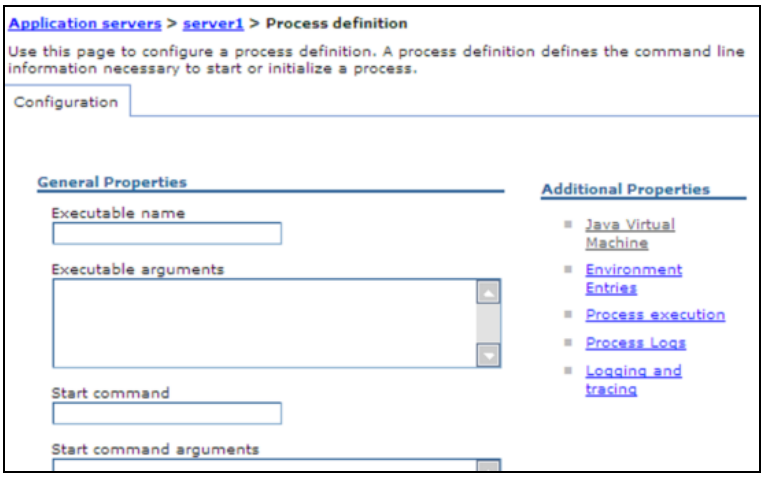

Figure 3.54: Navigating to the Java Virtual Machine properties of the target WebSphere server

13. In the **Generic JVM arguments** text box of the **Configuration** section tab page that appears, add the following line:

    -Djavax.management.builder.initial= -Dcom.sun.management.jmxremote

14. Then, click the **Apply** and **OK** buttons.
15. Finally, stop and start the target WebSphere server.

3.7 Managing IBM WebSphere Application Server 6.0 and above

To achieve this, do the following:

1. Login to the eG administrative interface as admin.
2. Check if the WebSphere application server is already discovered. If already discovered, use the **COMPONENTS - MANAGE/UNMANAGE** page that appears on using the Infrastructure-> Components-> Manage/Unmanage menu to manage it. Otherwise, change the port information if needed and run discovery to discover it or manually add the WebSphere Server. For adding, use the **COMPONENT** page that can be accessed through the menu sequence Infrastructure -> Components -> Add/Modify. While components manually added are automatically managed, proceed to manually manage the discovered components using the **COMPONENTS - MANAGE/UNMANAGE** page.
3. Now, if you proceed to log out of the administrative interface, a list of unconfigured tests (see Figure 3.55) will appear prompting you to configure tests for the WebSphere Server.

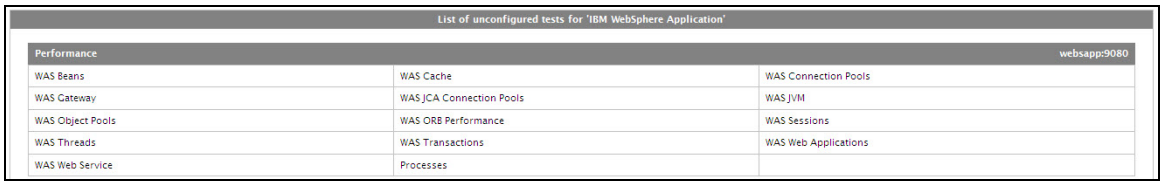

Figure 3.55: A page displaying WebSphere tests to be configured

4. Click on, say, **WAS Beans** test to move to the page that facilitates test configuration (see Figure 3.56). This test reports the performance metrics that determines the functioning of the individual EJBs on the WAS server.

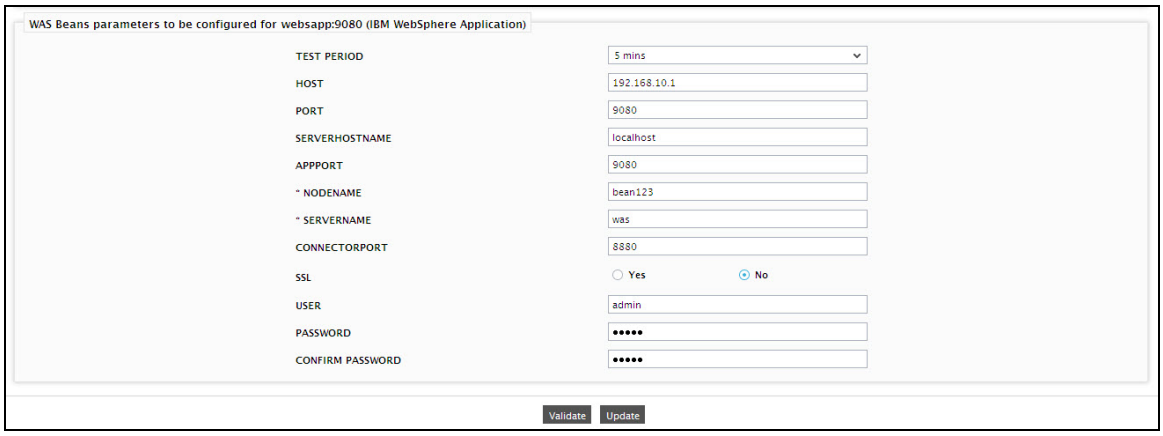

Figure 3.56: Specifying the test parameters for a WebSphere application server

5. Here, select the WebSphere server for which the test is to be configured, click on the **Configure** button, and provide the following information:

   - **TEST PERIOD** - how often should the test be executed
   - **HOST** – The IP address of the WebSphere application server
   - **PORT** – The port number of the WebSphere application server
   - **SERVERHOSTNAME** - Specify the host name of the application server instance being monitored.
   - **APPPORT** – Specify the port number to be used for accessing the egurkha application that has been deployed on the server.
   - **NODENAME** - Specify the node name of the server instance being monitored. To know the node name, do the following:

- Login to the WebSphere Administrative Console.
- Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application servers** link within (see Figure 3.57).

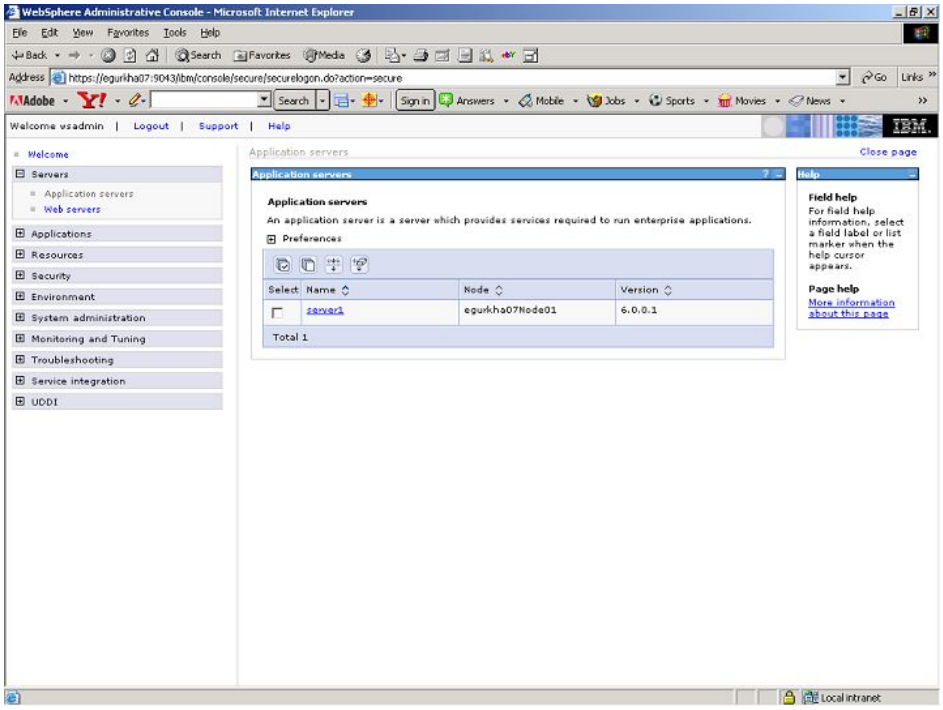

Figure 3.57: The WebSphere Administrative Console displaying the list of application server instances

- A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the **Node** name that corresponds to the application server instance being monitored, and specify that name against the nodename parameter.

  Note:

  As websphere is case sensitive, make sure the **NODE NAME** is specified correctly.

- **SERVERNAME** – Provide the name of the server instance being monitored in the SERVERNAME text box. To know the server name, do the following:

  - Login to the WebSphere Administrative Console.
  - Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application servers** link within (see Figure 3.58).

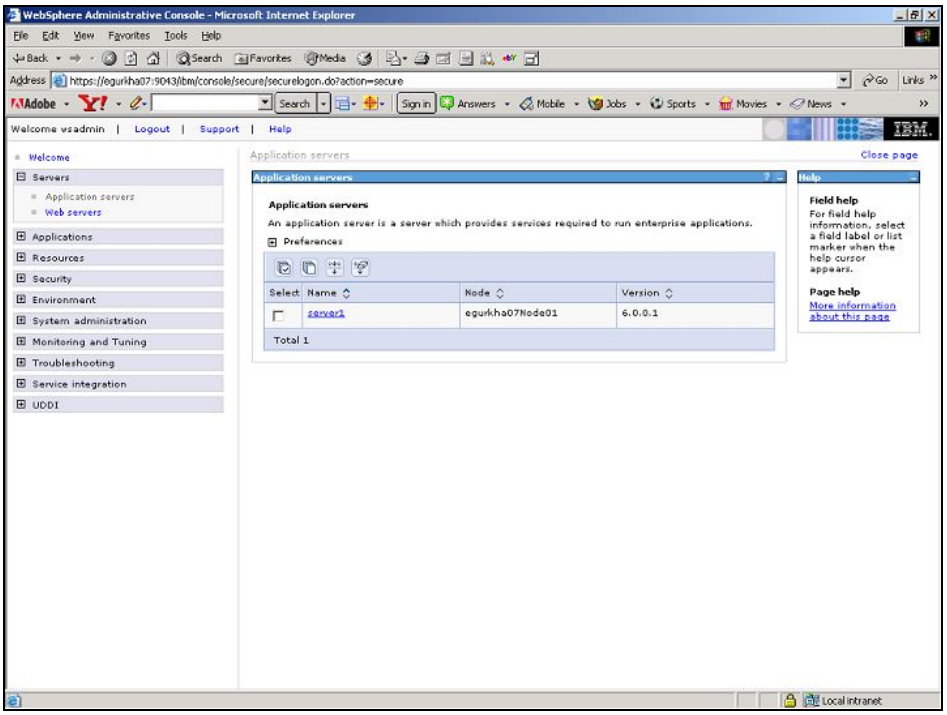

Figure 3.58: The instances of a WebSphere server

- A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane (see Figure 3.58).
- One of the displayed server instance **Names** can be specified as the value of the servername parameter.

If the server instance being monitored is part of a WebSphere cluster, then you need to provide the host name that corresponds to the connector port of the *Deployment Manager* of the cluster as the **SERVERNAME**. To determine the **SERVERNAME** in this case, do the following:

- Connect to the WebSphere Administrative console using the URL: http://<DeploymentManagerIP>:<DeploymentManagerPort> .
- Login to the WebSphere Administrative console. Figure 3.59 then appears.

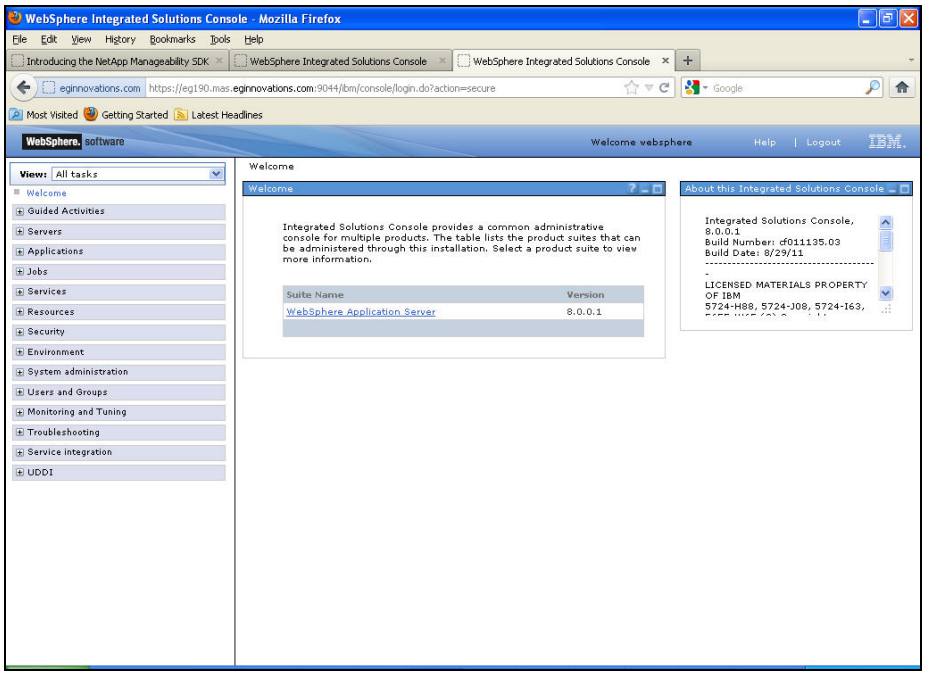

Figure 3.59: The Administrative console of the WebSphere Deployment Manager

- Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within.

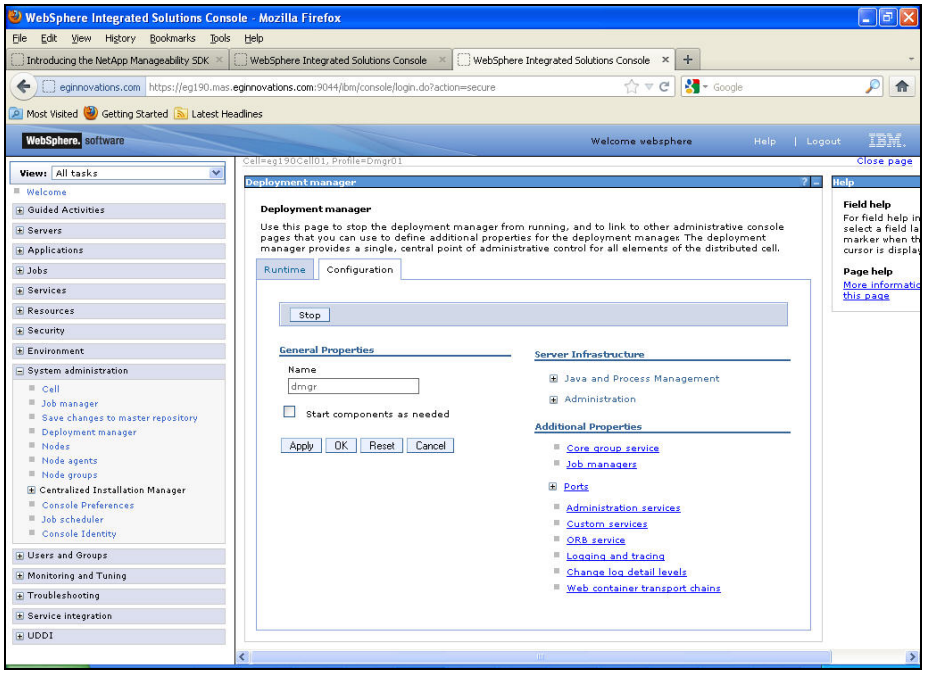

Figure 3.60: The Deployment Manager Configuration

- In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager** (see Figure 3.121). In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node.
- A list of ports will then appear. Click on the **Details** button alongside the ports list (see Figure 3.61).

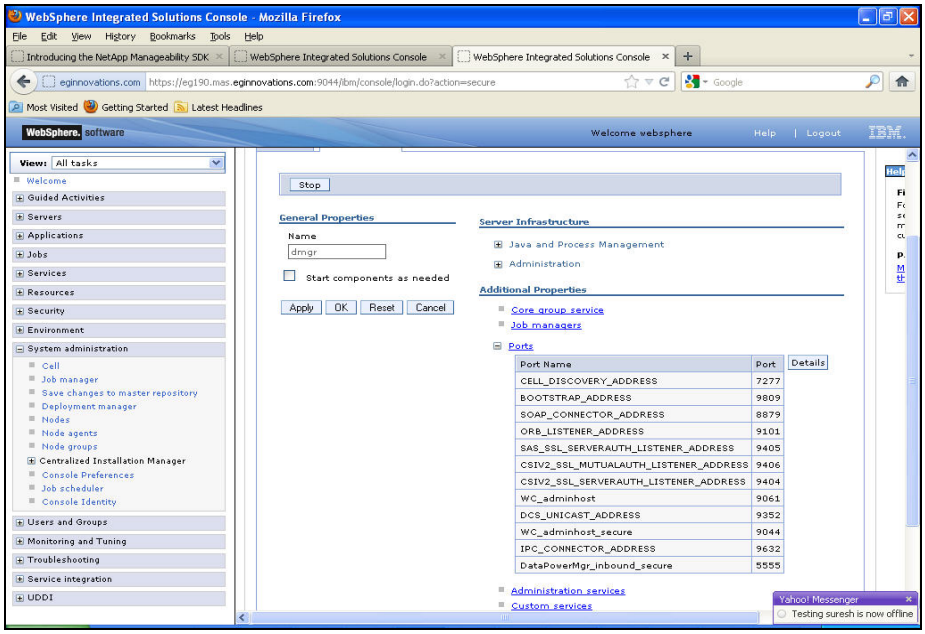

Figure 3.61: The list of ports on the Deployment Manager

- If the SOAP port has been set as the connector port in your environment, then scroll down the page that appears next until you view the **Port Name**, **SOAP_CONNECTOR_ADDRESS**. Make note of the **Host** name that corresponds to this port name. If the RMI port is the connector port in your environment, then make note of the **Host** name that corresponds to the **Port** name, **BOOTSTRAP_ADDRESS** (see Figure 3.62).

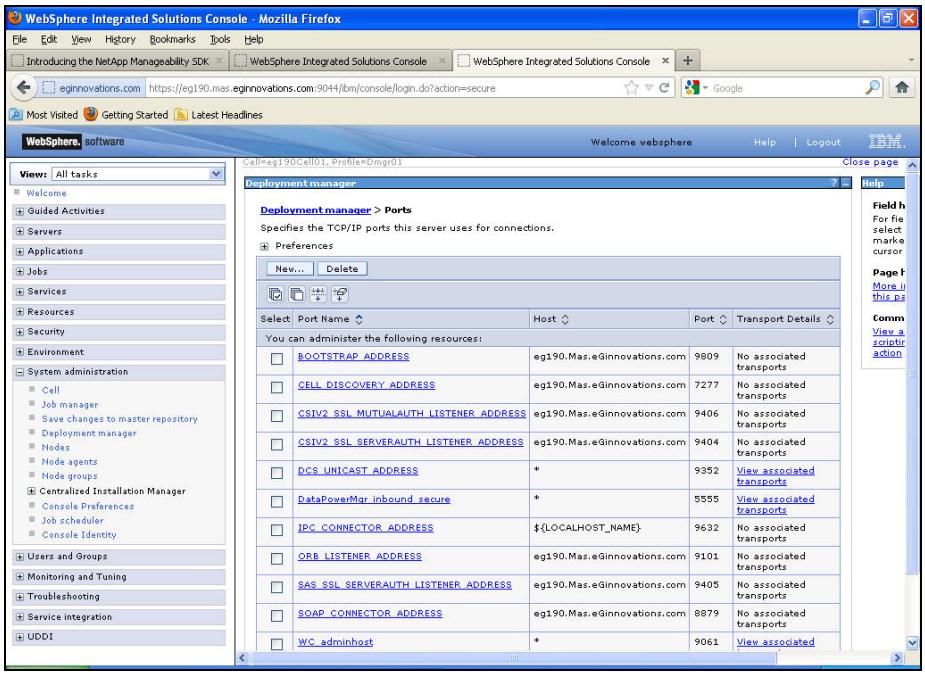

Figure 3.62: Host name that corresponds to the Connector Port

- Specify this **Host** name as the servername.

- **CONNECTORPORT** - The applications that are deployed on a server instance use the **CONNECTORPORT** for all internal communications with the application server. The connector port can be a SOAP port or an RMI port. The default connector port however, is the SOAP port. To know the connector port number, do the following:

  - Login to the WebSphere Administrative Console.
  - Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application servers** link within (see Figure 3.63).

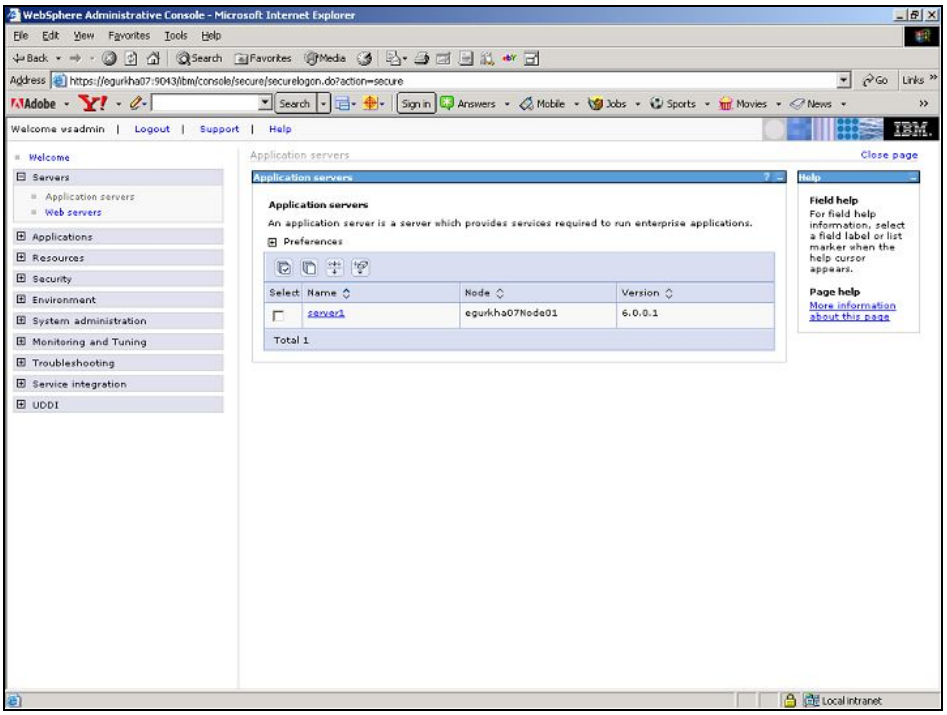

Figure 3.63: Opening the Deployment Manager node

- A list of application server instances and their corresponding node names will then appear in the right pane of the console. In the right pane, click on the server name link that corresponds to the server instance that is being monitored. Figure 3.64 revealing the **Configuration** of the server instance clicked on, then appears.

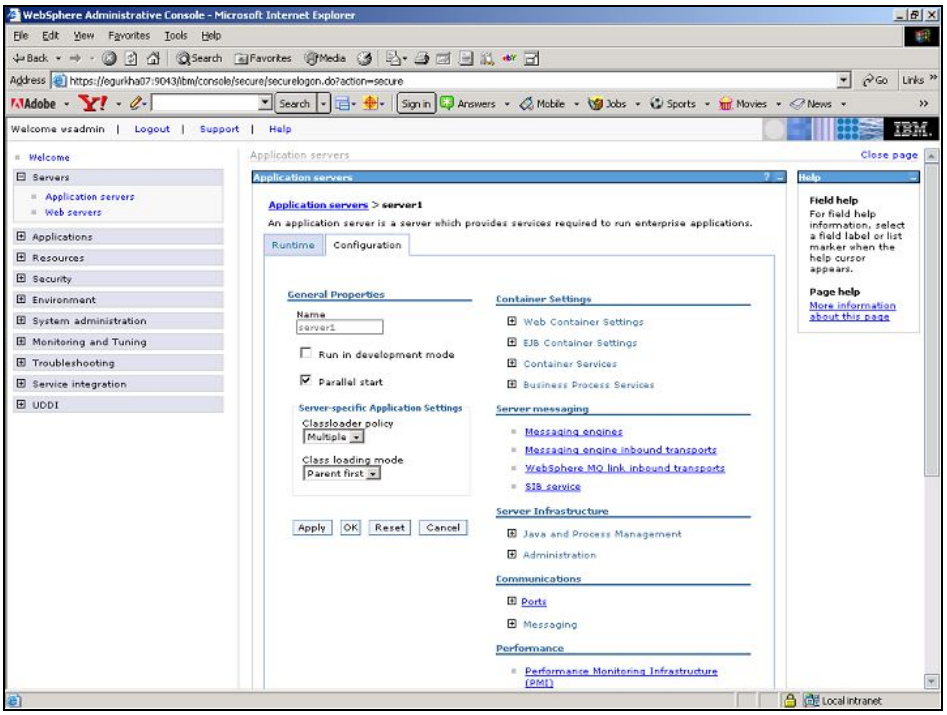

Figure 3.64: Configuration of the server instance clicked on

- Scroll down the **Configuration** tab page to view the **Communications** section (see Figure 3.65).

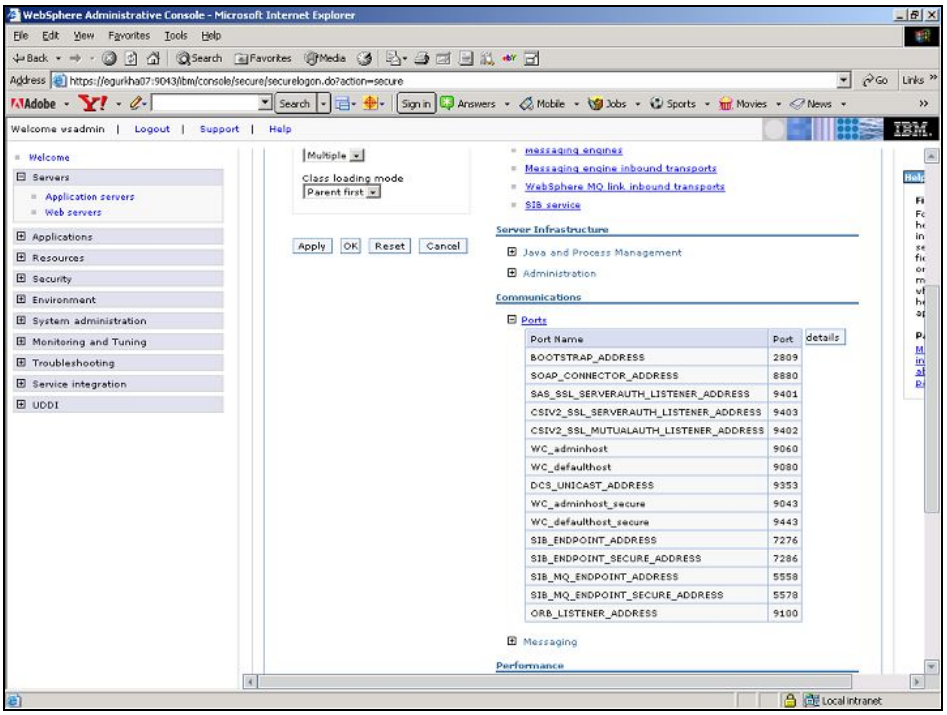

Figure 3.65: Scrolling down to view the Communications section

- Expand the **Ports** link in the **Configuration** section (as depicted by Figure 3.65) to view a list of ports. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the CONNECTORPORT. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the CONNECTORPORT.

If the server instance being monitored is part of a WebSphere cluster, then you need to provide the connector port of the *Deployment Manager* of the cluster as the **CONNECTORPORT**. To determine the **CONNECTORPORT** in this case, do the following:

- Connect to the WebSphere Administrative console using the URL: http://<DeploymentManagerIP>:<DeploymentManagerPort > or https://<DeploymentManagerIP>:<DeploymentManagerPort> .
- Login to the WebSphere Administrative console. 3.7 then appears.

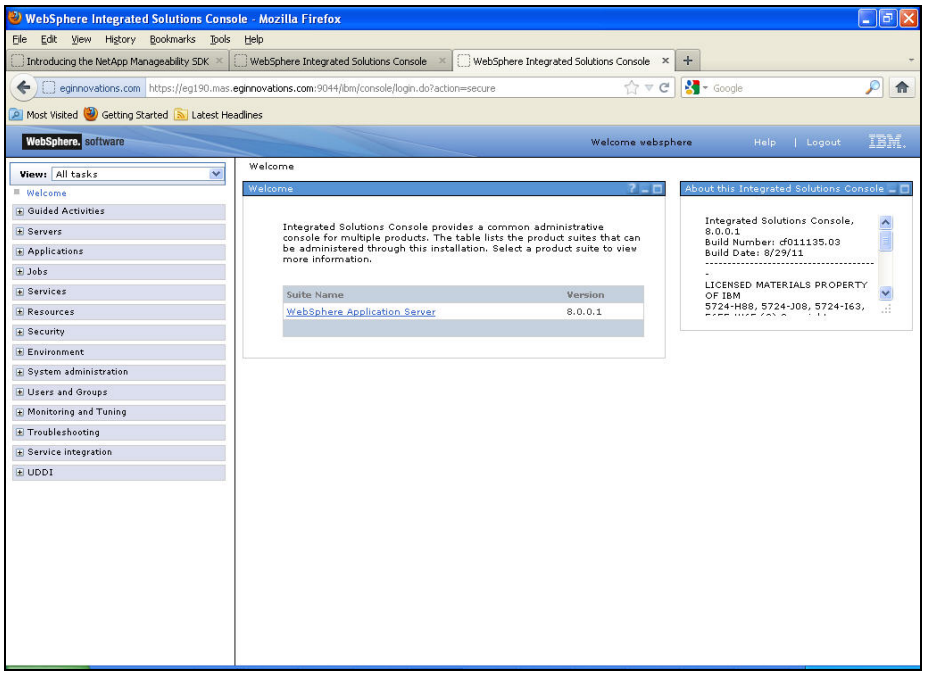

Figure 3.66: The Administrative console of the WebSphere Deployment Manager

- Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within.

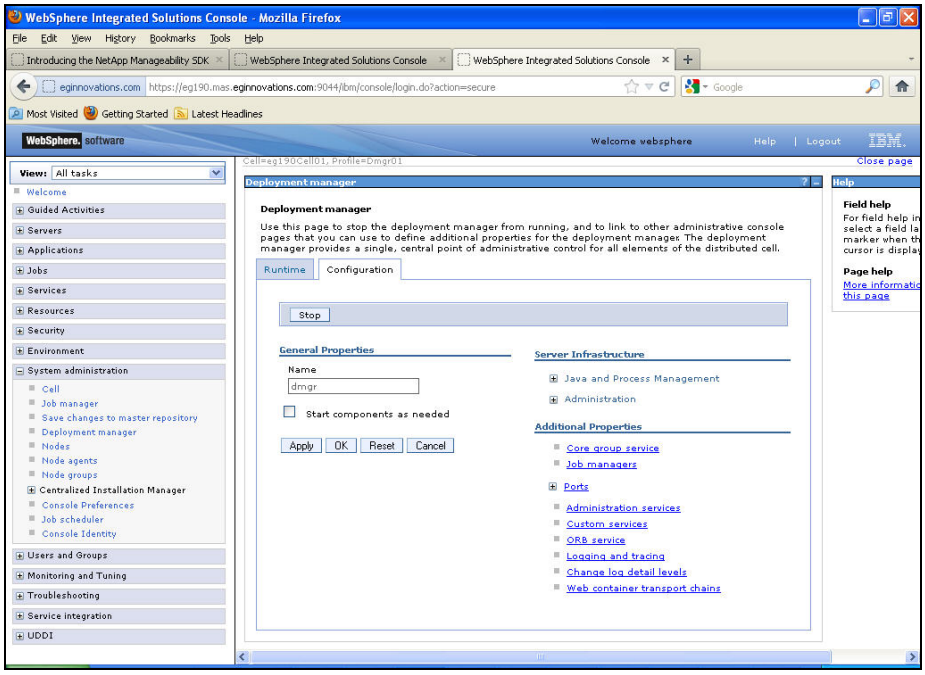

Figure 3.67: The Deployment Manager Configuration

- In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager** (see Figure 3.121). In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node.
- A list of ports will then appear. If the default connector port is in use, then the port number displayed against **SOAP_ CONNECTOR_ ADDRESS** should be specified as the CONNECTORPORT. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ ADDRESS** as the CONNECTORPORT.

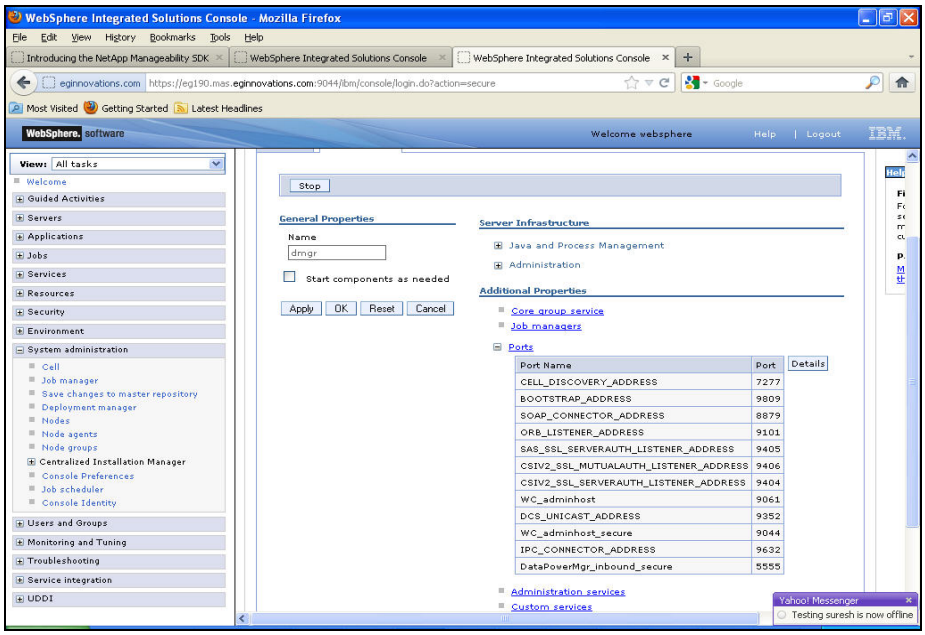

Figure 3.68: The list of ports on the Deployment Manager

- **SSL** – Select Yes if **SSL** (Secured Socket Layer) is to be used to connect to the WebSphere server, and No if it is not.
- **USER** - If security has been enabled for the WebSphere server being monitored, then provide a valid **USER** name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the **USER** text box should contain the default value 'none'.
- **PASSWORD** – If security has been enabled for the WebSphere server being monitored, then provide the password that corresponds to the specified user name. If the WebSphere server does not require any authentication, then leave the password text box with its default setting.
- **CONFIRM PASSWORD** - If security has been confirm the specified password by retyping it in the confirm password text box. If the WebSphere server does not require any authentication, then leave the confirm password text box with its default setting.

- **DETAILED DIAGNOSIS** - To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option.

  The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled:

  - The eG manager license should allow the detailed diagnosis capability
  - Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

6. Then, click on the **Update** button to save the changes.
7. Try to signout again; this time, you will be prompted to configure **Processes** test.
8. So, click on **Processes** test and proceed to configure it.

   Note:

   For **Processes** Test configuring procedure, refer to Configuring and Monitoring Mail Servers document.

9. Remember that the process pattern has to be specified in the following format:

   PROCESSNAME:PROCESSPATTERN

   For example, the process pattern can be specified as below:

   WebSphere:/opt/WebSphere/AppServer/java/jre/bin/exe/java*- Xmx128m*- Xminf0.15*- Xmaxf0.25*–X**

10. Use the context sensitive eG help to understand more about how this process pattern is coined.
11. Upon completion of Processes test configuration, signout of the administrative interface.
12. The WebSphere Application server is now ready to be monitored by eG.

Chapter 4: Monitoring the WebSphere Application Server Version 4/5.x

To obtain statistics specific to a WebSphere Application Server 4/5.x, an eG agent relies on the Performance Monitoring Interface (PMI) API supported by the WebSphere server. Through eG Enterprise's administrative interface, the webserver port number through which the WebSphere servers' applications can be accessed should be specified.

The layer model that the eG Enterprise suite uses for monitoring this WebSphere server is shown in Figure 4.1.

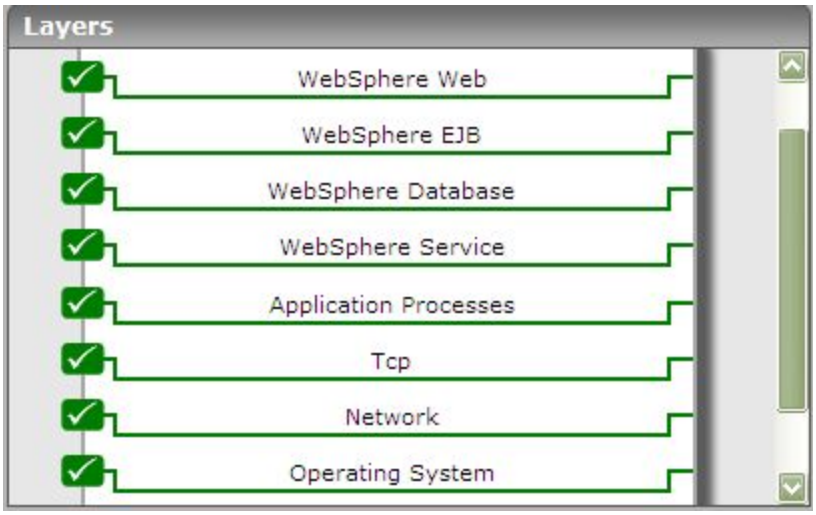

Figure 4.1: Layer model for a WebSphere application server – 4/5.x

The details pertaining to the **Operating System**, *N***etwork**, *T***cp**, and **Application Processes** layer have been discussed in the *Monitoring Unix and Windows Servers* document. Above the **Application Processes** layer is the **WebSphere Service** layer.

Note:

The WebSphere application server uses a specific Java-based application process to execute the class com.ibm.ejs.sm.server.AdminServer. The Processes test parameters for WebSphere application servers should be configured to monitor this process.

4.1 The WebSphere Service Layer

This layer tracks the health of the WebSphere service. The status of the **WebSphere Service** layer is determined by the results of the tests depicted by Figure 4.2.

Figure 4.2: Tests mapping to the WebSphere Service layer

4.1.1 WebSphere Jvm Test

This test monitors the usage of the WebSphere JVM heap.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each WebSphere application server being monitored

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WebServerPort	The port number through which the WebSphere applications can be accessed. (For IBMHTTPServer this information can be found in the *httpd.conf* file located in **$<INSTALLDIR>/conf** directory.)
SSL	Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.
ServerHostName	Specify the node name of the server instance being monitored. To know the node name, do the following:

Parameters	Description
	• Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the node name that corresponds to the application server instance being monitored, and specify that name against the ServerHostName parameter.
NDManager	The NDManager parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In the case of situation (a), in the NDManager text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the Node Manager, do the following: • Login to the Administrative Console of the node manager using the URL http://<IPoftheWebSphereAppServer'sNodeManager>:<Adminport>/admin/. • Using the tree-structure in the left pane of the Administrative Console that appears, drill down to the **Deployment Manager** node within **System Administration**. • Select the **Configuration** tab that appears in the right pane, and scroll down to the **End Points** link in the **Additional Properties** section. • Once you locate the **End Points** link, click on it, and in the page that appears, click on the **SOAP_CONNECTOR_ADDRESS** link. • In the subsequent page, a fully qualified domain name will be displayed against Host. This is the name that should be specified as the host name of the node manager in the NDManager text box. In the case of situation (b), enter the ServerHostName itself as the NDManager. If both conditions (a) and (b) do not apply, then specify *none* here.
ConnectorPort	The ConnectorPort parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or,

Parameters	Description
	b. If the WebSphere server being monitored is one of many instances of the server running on the same host In case of situation (a), the ConnectorPort parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the following: ◦ From the Node manager's host, open the <WEBSPHERE_ INSTALL_ DIR>\DeploymentManager\properties\wsadmin.properties file. ◦ Two entries that read as follows will be present in the wsadmin.properties file: *com.ibm.ws.scripting.port=<port_number>* *#com.ibm.ws.scripting.port=<port_number>* Both the entries will have port numbers against them. However, the uncommented entry (the entry without the #) is the one that denotes an active port. Therefore, specify the corresponding port number as the ConnectorPort. In case of situation (b), do the following to know the ConnectorPort: • Login to the WebSphere Administrative Console. • Expand the Servers node in the tree-structure in the left pane of the console, and click on the Application Servers link. A list of application server instances will then appear in the right pane. • Click on the server instance being monitored. Doing so invokes the **Configuration** of the chosen server instance appears. • Scroll down the **Configuration** to view the **End Points** link. Once you locate the **End Points** link, click on it. • In the page that appears next, click on the **SOAP_CONNECTOR_ADDRESS** link. • The Port displayed in that page should be used as the ConnectorPort in situation (b). If both (a) and (b) do not apply, specify *none* against ConnectorPort.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid User name to login to the WebSphere server. If the WebSphere server does not

Parameters	Description
	require any authentication, then the User text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.
EncryptPass	By default, this flag is set to **Yes**, indicating that the Password of the WebSphere server is encrypted by default. To disable password encryption, select the **No** option.
ServerName	Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, first, connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node. A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane. One of the displayed server instances can be specified as the value of the ServerName parameter.
Timeout	In the Timeout text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default Timeout period is 60 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
JVM total memory	Indicates the total heap size of the JVM.	MB	
JVM used memory	Indicates the amount of memory that has been utilized by the JVM.	MB	A high value of this measure indicates a heavy workload on the WebSphere application server. In such a case, you might want to consider increasing the JVM heap size.

Measurement	Description	Measurement Unit	Interpretation
JVM free memory	Indicates the amount of memory currently available in the JVM.	MB	A very low value of this measure is indicative of excessive memory utilization in the JVM.

4.1.2 WebSphere Test

Since the WebSphere application server uses Java extensively, the performance of the Java Virtual Memory (JVM) can impact the WebSphere server's performance. This test measures the status of the JVM and the server's availability.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each WebSphere application server being monitored

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WebServerPort	The port number through which the WebSphere applications can be accessed. (For IBMHTTPServer this information can be found in the *httpd.conf* file located in **$<INSTALLDIR>/conf** directory.)
SSL	Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.
ServerHostName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then

Parameters	Description
	appear in the right pane of the console. From this list, you can figure out the node name that corresponds to the application server instance being monitored, and specify that name against the ServerHostName parameter.
NDManager	The NDManager parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In the case of situation (a), in the NDManager text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the Node Manager, do the following: • Login to the Administrative Console of the node manager using the URL http://<IPoftheWebSphereAppServer'sNodeManager>:<Adminport>/admin/. • Using the tree-structure in the left pane of the Administrative Console that appears, drill down to the **Deployment Manager** node within **System Administration**. • Select the **Configuration** tab that appears in the right pane, and scroll down to the **End Points** link in the **Additional Properties** section. • Once you locate the **End Points** link, click on it, and in the page that appears, click on the **SOAP_CONNECTOR_ADDRESS** link. • In the subsequent page, a fully qualified domain name will be displayed against Host. This is the name that should be specified as the host name of the node manager in the NDManager text box. In the case of situation (b), enter the ServerHostName itself as the NDManager. If both conditions (a) and (b) do not apply, then specify *none* here.
ConnectorPort	The ConnectorPort parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In case of situation (a), the ConnectorPort parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the

Parameters	Description
	following: ◦ From the Node manager's host, open the <WEBSPHERE_ INSTALL_ DIR>\DeploymentManager\properties\wsadmin.properties file. ◦ Two entries that read as follows will be present in the wsadmin.properties file: *com.ibm.ws.scripting.port=<port_number>* *#com.ibm.ws.scripting.port=<port_number>* Both the entries will have port numbers against them. However, the uncommented entry (the entry without the #) is the one that denotes an active port. Therefore, specify the corresponding port number as the ConnectorPort. In case of situation (b), do the following to know the ConnectorPort: • Login to the WebSphere Administrative Console. • Expand the Servers node in the tree-structure in the left pane of the console, and click on the Application Servers link. A list of application server instances will then appear in the right pane. • Click on the server instance being monitored. Doing so invokes the **Configuration** of the chosen server instance appears. • Scroll down the **Configuration** to view the **End Points** link. Once you locate the **End Points** link, click on it. • In the page that appears next, click on the **SOAP_CONNECTOR_ADDRESS** link. • The Port displayed in that page should be used as the ConnectorPort in situation (b). If both (a) and (b) do not apply, specify *none* against ConnectorPort.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid User name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the User text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.

Parameters	Description
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.
EncryptPass	By default, this flag is set to **Yes**, indicating that the Password of the WebSphere server is encrypted by default. To disable password encryption, select the **No** option.
ServerName	Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, first, connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node. A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane. One of the displayed server instances can be specified as the value of the ServerName parameter.
Timeout	In the Timeout text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default Timeout period is 60 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Heap usage	Indicates the total JVM heap used.	Percent	A high percentage of the heap usage indicates either the applications running on this server might be utilizing/leaking too much memory or the server has reached the maximum memory capacity.
Availability	Indicates the availability of the server.	Percent	A value of 100 for this measure indicates that the server is available. Any other value indicates that the server is not running.

4.1.3 WebSphere Thread Pools Test

To optimize performance and at the same time to support concurrent accesses from users, the application server uses thread pools. It is critical to monitor a WebSphere server's thread pools on an ongoing basis. This test does just that.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each thread pool that exists on a WebSphere application server

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WebServerPort	The port number through which the WebSphere applications can be accessed. (For IBMHTTPServer this information can be found in the *httpd.conf* file located in **$<INSTALLDIR>/conf** directory.)
SSL	Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.
ServerHostName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the node name that corresponds to the application server instance being monitored, and specify that name against the ServerHostName parameter.
NDManager	The NDManager parameter is applicable only under the following circumstances:

Parameters	Description
	a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In the case of situation (a), in the NDManager text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the Node Manager, do the following: • Login to the Administrative Console of the node manager using the URL http://<IPoftheWebSphereAppServer'sNodeManager>:<Adminport>/admin/. • Using the tree-structure in the left pane of the Administrative Console that appears, drill down to the **Deployment Manager** node within **System Administration**. • Select the **Configuration** tab that appears in the right pane, and scroll down to the **End Points** link in the **Additional Properties** section. • Once you locate the **End Points** link, click on it, and in the page that appears, click on the **SOAP_CONNECTOR_ADDRESS** link. • In the subsequent page, a fully qualified domain name will be displayed against Host. This is the name that should be specified as the host name of the node manager in the NDManager text box. In the case of situation (b), enter the ServerHostName itself as the NDManager. If both conditions (a) and (b) do not apply, then specify *none* here.
ConnectorPort	The ConnectorPort parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In case of situation (a), the ConnectorPort parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the following: ◦ From the Node manager's host, open the <WEBSPHERE_ INSTALL_ DIR>\DeploymentManager\properties\wsadmin.properties file. ◦ Two entries that read as follows will be present in the wsadmin.properties file:

Parameters	Description
	com.ibm.ws.scripting.port=<port_number> *#com.ibm.ws.scripting.port=<port_number>* Both the entries will have port numbers against them. However, the uncommented entry (the entry without the #) is the one that denotes an active port. Therefore, specify the corresponding port number as the ConnectorPort. In case of situation (b), do the following to know the ConnectorPort: • Login to the WebSphere Administrative Console. • Expand the Servers node in the tree-structure in the left pane of the console, and click on the Application Servers link. A list of application server instances will then appear in the right pane. • Click on the server instance being monitored. Doing so invokes the **Configuration** of the chosen server instance appears. • Scroll down the **Configuration** to view the **End Points** link. Once you locate the **End Points** link, click on it. • In the page that appears next, click on the **SOAP_CONNECTOR_ADDRESS** link. • The Port displayed in that page should be used as the ConnectorPort in situation (b). If both (a) and (b) do not apply, specify *none* against ConnectorPort.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid User name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the User text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.
EncryptPass	By default, this flag is set to **Yes**, indicating that the Password of the WebSphere server is encrypted by default. To disable password encryption, select the **No** option.

Parameters	Description
ServerName	Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, first, connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node. A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane. One of the displayed server instances can be specified as the value of the ServerName parameter.
Timeout	In the Timeout text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default Timeout period is 60 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Threads created	This measure indicates the rate at which the threads are being created in the pool.	Threads/Sec	A sudden increase in the value of this measure directly relates to an increase in the activity happening in this application.
Threads destroyed	This measure indicates the rate at which the threads are being destroyed in the pool.	Threads/Sec	A decrease in the value of this measure indicates that the threads are being active for a long period of time, which might indicate anomalies within the application.
Active threads	Indicates the average number of active threads in the pool.	Number	A high value for this measure is indicative of a high load on this application and combined with the creation and destroy rates might give insights of the application pattern. This measure is also useful for determining usage trends. For example, it can show the time of

Measurement	Description	Measurement Unit	Interpretation
			day and the day of the week in which you usually reach peak thread count. In addition, the creation of too many threads can result in out of memory errors or thrashing. By watching this metric, you can reduce excessive memory consumption before it's too late.
Pool size	Indicates the average number of threads in the pool.	Number	If the pool size is high and the number of active threads is low, it signifies that the threads are not being destroyed immediately after use.
Percent maxed	This indicates the average percent of time the number of threads in the pool reached or exceeded the configured maximum number.	Percent	A high value for this measure indicates that the number of threads in the pool needs to be increased for effective operation of this application. Alternatively, you may also consider changing the maximum number of threads that a pool can contain However, exercise caution when altering the maximum thread count, as a very high thread count can cause the app to slowdown from excessive memory usage. Likewise, if the maximum thread count is set too low, it will cause requests to block or timeout. You can use this measure to see how often you are reaching the maximum thread count in a pool. This can be used to tune other properties – such as the amount of time before an idle thread is destroyed and the frequency of when new threads are created.

4.2 The WebSphere Database Layer

This layer tracks the statistics of the database access (connection pools) of the applications hosted on a WebSphere application server with the help of the WebSphereJdbc test (see Figure 4.3).

Figure 4.3: Tests mapping to the WebSphere Database layer

4.2.1 WebSphere JDBC Test

Rather than opening and closing connections for individual database accesses, the WebSphere application server creates pools of database connections that are reused. To understand the performance of a WebSphere application server, it is critical to monitor the usage of the connection pools by the server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each database connection pool that exists on the WebSphere application server being monitored

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WebServerPort	The port number through which the WebSphere applications can be accessed.
SSL	Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.

Parameters	Description
ServerHostName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the node name that corresponds to the application server instance being monitored, and specify that name against the ServerHostName parameter.
NDManager	The NDManager parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In the case of situation (a), in the NDManager text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the Node Manager, do the following: • Login to the Administrative Console of the node manager using the URL http://<IPoftheWebSphereAppServer'sNodeManager>:<Adminport>/admin/. • Using the tree-structure in the left pane of the Administrative Console that appears, drill down to the **Deployment Manager** node within **System Administration**. • Select the **Configuration** tab that appears in the right pane, and scroll down to the **End Points** link in the **Additional Properties** section. • Once you locate the **End Points** link, click on it, and in the page that appears, click on the **SOAP_CONNECTOR_ADDRESS** link. • In the subsequent page, a fully qualified domain name will be displayed against Host. This is the name that should be specified as the host name of the node manager in the NDManager text box. In the case of situation (b), enter the ServerHostName itself as the NDManager. If both conditions (a) and (b) do not apply, then specify *none* here.

Parameters	Description
ConnectorPort	The ConnectorPort parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In case of situation (a), the ConnectorPort parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the following: ◦ From the Node manager's host, open the <WEBSPHERE_INSTALL_DIR>\DeploymentManager\properties\wsadmin.properties file. ◦ Two entries that read as follows will be present in the wsadmin.properties file: *com.ibm.ws.scripting.port=<port_number>* *#com.ibm.ws.scripting.port=<port_number>* Both the entries will have port numbers against them. However, the uncommented entry (the entry without the #) is the one that denotes an active port. Therefore, specify the corresponding port number as the ConnectorPort. In case of situation (b), do the following to know the ConnectorPort: • Login to the WebSphere Administrative Console. • Expand the Servers node in the tree-structure in the left pane of the console, and click on the Application Servers link. A list of application server instances will then appear in the right pane. • Click on the server instance being monitored. Doing so invokes the **Configuration** of the chosen server instance appears. • Scroll down the **Configuration** to view the **End Points** link. Once you locate the **End Points** link, click on it. • In the page that appears next, click on the **SOAP_CONNECTOR_ADDRESS** link. • The Port displayed in that page should be used as the ConnectorPort in situation (b). If both (a) and (b) do not apply, specify *none* against ConnectorPort.

Parameters	Description
User	If security has been enabled for the WebSphere server being monitored, then provide a valid User name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the User text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.
EncryptPass	By default, this flag is set to **Yes**, indicating that the Password of the WebSphere server is encrypted by default. To disable password encryption, select the **No** option.
ServerName	Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, first, connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node. A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane. One of the displayed server instances can be specified as the value of the ServerName parameter.
Timeout	In the Timeout text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default Timeout period is 60 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Database connection rate	Indicates the rate at which the connections are created in the pool.	Conns/Sec	A sudden increase in the value of this measure directly relates to an increase in the database activity happening in this application.
Connection destroy rate	Indicates the rate at which the connections are released from the pool.	Conns/Sec	A decrease in the value of this measure points to an unusual performance of the application. The

Measurement	Description	Measurement Unit	Interpretation
			application might not be releasing the connections.
Connection return rate	This measure indicates the rate at which the connections are being returned to the pool.	Conns/Sec	A low value of this measure might be due to some applications that are not releasing the connections to the pool.
Allocate rate	Indicates the rate at which the connections are allocated from the pool.	Conns/Sec	A high value for this measure indicates more activity happening on the server. An unusually high value for this measure might result in the pool running out of connections, if the Return_rate is not concurrent.
Pool size	Indicates the number of connections in the pool. It indicates the active pool size.	Number	A growing pool size may be due to some of the applications not releasing connections to the pool.
Concurrent waiters	Indicates the average number of threads waiting for a connection.	Number	A high value indicates a bottleneck on the application server. This may be caused due to unreleased connections.
Avg wait time	Indicates the average time that a client waited to be granted a connection.	Secs	An increase in this measure reflects a server bottleneck.
Faults	Indicates the rate at which connection pool timeouts occur.	Faults/Sec	A high value is indicative of a bottleneck on the server.
Usage	Indicates the average percentage of connections of the pool in use	Percent	This measure should not reach the maximum of 100% for optimal performance of the application server. To reduce this value, try increasing the pool size.
Percent maxed	Average percentage of the time that all connections are in use.	Percent	A high value indicates that the pool size needs to be increased for effective operation of this application.
Prepared statement	Indicates the rate at which	Stmts/Sec	A high value indicates a bottleneck in

Measurement	Description	Measurement Unit	Interpretation
cache discard rate	the prepared statements are discarded from the cache.		the application. It is a sign of inefficient performance of the application.

4.3 The WebSphere EJB Layer

The Ws Beans test shown in Figure 4.4 determines the status of the **WebSphere EJB** layer. This layer determines the status of specific Enterprise Java Bean (abbreviated as EJB in this manual) components hosted on a WebSphere server. The details of these tests are provided below:

Figure 4.4: Tests mapping to the WebSphere EJB layer

4.3.1 Ws Beans Test

Enterprise Java Beans (EJB) forms the critical components of any business application. WebSphere application server creates instances corresponding to the EJB. This test reports the performance metrics that determines the functioning of the EJB groups. Use the **Click here** hyperlink in the test configuration page to configure the EJB groups that need to be monitored by the eG Enterprise suite. **By default, the eG Enterprise system monitors only those EJBs that are part of a group.**

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every EJB group configured

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WebServerPort	The port number through which the WebSphere applications can be accessed.
SSL	Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.
ServerHostName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the node name that corresponds to the application server instance being monitored, and specify that name against the ServerHostName parameter.
NDManager	The NDManager parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In the case of situation (a), in the NDManager text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the Node Manager, do the following: • Login to the Administrative Console of the node manager using the URL http://<IPoftheWebSphereAppServer'sNodeManager>:<Adminport>/admin/. • Using the tree-structure in the left pane of the Administrative Console that appears, drill down to the **Deployment Manager** node within **System Administration**. • Select the **Configuration** tab that appears in the right pane, and scroll down to

Parameters	Description
	the **End Points** link in the **Additional Properties** section. • Once you locate the **End Points** link, click on it, and in the page that appears, click on the **SOAP_CONNECTOR_ADDRESS** link. • In the subsequent page, a fully qualified domain name will be displayed against Host. This is the name that should be specified as the host name of the node manager in the NDManager text box. In the case of situation (b), enter the ServerHostName itself as the NDManager. If both conditions (a) and (b) do not apply, then specify *none* here.
ConnectorPort	The ConnectorPort parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In case of situation (a), the ConnectorPort parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the following: ◦ From the Node manager's host, open the <WEBSPHERE_INSTALL_DIR>\DeploymentManager\properties\wsadmin.properties file. ◦ Two entries that read as follows will be present in the wsadmin.properties file: *com.ibm.ws.scripting.port=<port_number>* *#com.ibm.ws.scripting.port=<port_number>* Both the entries will have port numbers against them. However, the uncommented entry (the entry without the #) is the one that denotes an active port. Therefore, specify the corresponding port number as the ConnectorPort. In case of situation (b), do the following to know the ConnectorPort: • Login to the WebSphere Administrative Console. • Expand the Servers node in the tree-structure in the left pane of the console, and click on the Application Servers link. A list of application server instances will then appear in the right pane. • Click on the server instance being monitored. Doing so invokes the **Configuration** of

Parameters	Description
	the chosen server instance appears. • Scroll down the **Configuration** to view the **End Points** link. Once you locate the **End Points** link, click on it. • In the page that appears next, click on the **SOAP_CONNECTOR_ADDRESS** link. • The Port displayed in that page should be used as the ConnectorPort in situation (b). If both (a) and (b) do not apply, specify *none* against ConnectorPort.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid User name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the User text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.
EncryptPass	By default, this flag is set to **Yes**, indicating that the Password of the WebSphere server is encrypted by default. To disable password encryption, select the **No** option.
ServerName	Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, first, connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node. A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane. One of the displayed server instances can be specified as the value of the ServerName parameter.
Timeout	In the Timeout text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default Timeout period is 60 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Bean instantiations	This measure indicates the rate at which the bean objects are instantiated.	Instances/Sec	A sudden increase in the rate of instantiations indicates a bottleneck on the server. It may be due to greater load on the server or there might be a loophole in the application.
Bean destroys	Indicates the rate at which the bean objects are destroyed.	Instances/Sec	A very low value indicates a bottleneck on the server. This might affect the performance of the application.
Bean method calls	This measure indicates the rate at which the methods are being processed by the server.	Methods/Sec	A high value indicates that the server is busy.
Avg method rate	This measure indicates the average response time of all methods of the remote interface for this bean.	Secs	A sudden increase in the value of this measure is a sign of overload on the server.
Avg bean creation time	Indicates the average method response time for creates.	Secs	This value should be low for optimal performance of the application server. The value may go high, if there are more objects in the pool, which is a sign of overload. This measure will not be available if the instrumentation level is set to H, instead of X.
Bean pool size	Indicates the average number of objects in the pool.	Number	A large pool size signifies that some of the objects are not being released by the application.

4.4 The WebSphere Web Layer

This layer tracks the health of all web applications and transactions that exist on the server. Figure 4.5 depicts the tests that map to this layer.

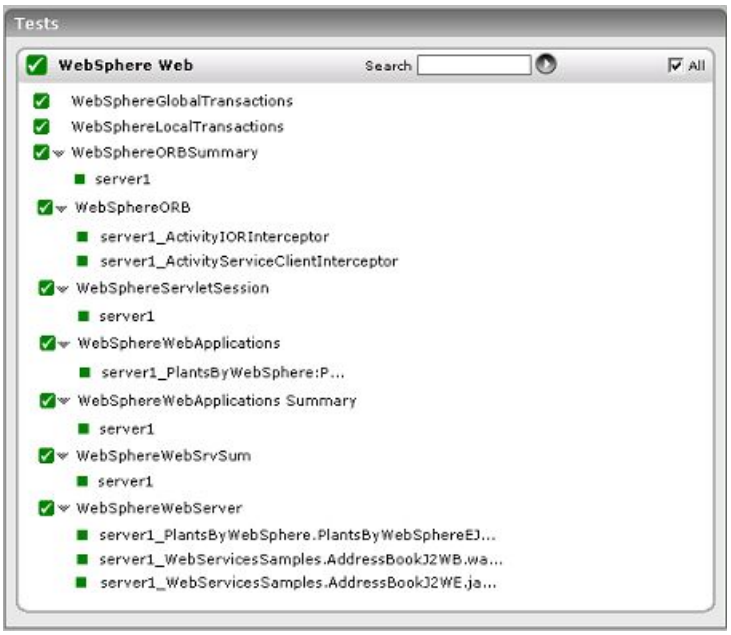

Figure 4.5: Tests mapping to the WebSphere Web layer

4.4.1 WebSphereGlobalTransactions Test

Transactions are the key functionality of the WebSphere application server. Global transactions are those that take place in more than one server. This test monitors the global transactions that are happening on different servers.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each WebSphere application server being monitored

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WebServerPort	The port number through which the WebSphere applications can be accessed.
SSL	Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.

Parameters	Description
ServerHostName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the node name that corresponds to the application server instance being monitored, and specify that name against the ServerHostName parameter.
NDManager	The NDManager parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In the case of situation (a), in the NDManager text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the Node Manager, do the following: • Login to the Administrative Console of the node manager using the URL http://<IPoftheWebSphereAppServer'sNodeManager>:<Adminport>/admin/. • Using the tree-structure in the left pane of the Administrative Console that appears, drill down to the **Deployment Manager** node within **System Administration**. • Select the **Configuration** tab that appears in the right pane, and scroll down to the **End Points** link in the **Additional Properties** section. • Once you locate the **End Points** link, click on it, and in the page that appears, click on the **SOAP_CONNECTOR_ADDRESS** link. • In the subsequent page, a fully qualified domain name will be displayed against Host. This is the name that should be specified as the host name of the node manager in the NDManager text box. In the case of situation (b), enter the ServerHostName itself as the NDManager. If both conditions (a) and (b) do not apply, then specify *none* here.

Parameters	Description
ConnectorPort	The ConnectorPort parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In case of situation (a), the ConnectorPort parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the following: ◦ From the Node manager's host, open the <WEBSPHERE_INSTALL_DIR>\DeploymentManager\properties\wsadmin.properties file. ◦ Two entries that read as follows will be present in the wsadmin.properties file: *com.ibm.ws.scripting.port=<port_number>* *#com.ibm.ws.scripting.port=<port_number>* Both the entries will have port numbers against them. However, the uncommented entry (the entry without the #) is the one that denotes an active port. Therefore, specify the corresponding port number as the ConnectorPort. In case of situation (b), do the following to know the ConnectorPort: • Login to the WebSphere Administrative Console. • Expand the Servers node in the tree-structure in the left pane of the console, and click on the Application Servers link. A list of application server instances will then appear in the right pane. • Click on the server instance being monitored. Doing so invokes the **Configuration** of the chosen server instance appears. • Scroll down the **Configuration** to view the **End Points** link. Once you locate the **End Points** link, click on it. • In the page that appears next, click on the **SOAP_CONNECTOR_ADDRESS** link. • The Port displayed in that page should be used as the ConnectorPort in situation (b). If both (a) and (b) do not apply, specify *none* against ConnectorPort.

Parameters	Description
User	If security has been enabled for the WebSphere server being monitored, then provide a valid User name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the User text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.
EncryptPass	By default, this flag is set to **Yes**, indicating that the Password of the WebSphere server is encrypted by default. To disable password encryption, select the **No** option.
ServerName	Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, first, connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node. A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane. One of the displayed server instances can be specified as the value of the ServerName parameter.
Timeout	In the Timeout text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default Timeout period is 60 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Global transactions begun	Indicates the rate at which global transactions are beginning to occur at the server.	Trans/Sec	A high value indicates an overload on the server.
Active global transactions	Indicates the average number of concurrently active global transactions.	Number	If the value of this measure is high, it signifies greater load on the server. This might increase the number of

Measurement	Description	Measurement Unit	Interpretation
			waiting local transactions.
Global transaction duration	Indicates the average duration of global transactions.	Secs	A high transaction duration value indicates increased load on the server.
Global commit duration	Indicates the average duration of commit for global transactions.	Secs	If the total time taken for committing a global transaction is high, then it indicates a bottleneck on the server.
Optimization rate	Indicates the rate at which global transactions being converted to single phase for optimization.	Trans/Sec	A sudden increase in this value might be caused due to one of the following reasons. 1. There might be more number of requests for the application. 2. There might be a bottleneck on the application.
Global transaction commits	Indicates the rate at which global transactions are being committed.	Transactions/Sec	If the rate of transactions that are being committed is very high, it signifies load on the server. It might be caused when some locked transactions are released suddenly.
Global transaction rollbacks	Indicates the rate at which the global transactions are being rolled back.	Trans/Sec	A high value indicates a problem with the application. Possible reasons could be the problem with the application or with some other dependent server (e.g. Database).
Global transaction timeouts	Indicates the rate at which the global transactions are being timed out.	Trans/Sec	Again, a rise in the value could be due to the problem with the application or with some other dependent server like the database.
Global transactions involved	Indicates the total number of global transactions involved at the server.	Number	A sudden increase in the value might be caused if the commit duration of a transaction goes low.
Global prepare duration	This measure indicates the average duration of prepare for the global transactions.	Secs	A high value indicates a bottleneck on the server.

4.4.2 WebSphere Local Transactions Test

Local transactions are those transactions that occur within the server. This test reports all the performance metrics associated with the local transactions.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each WebSphere application server being monitored

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WebServerPort	The port number through which the WebSphere applications can be accessed.
SSL	Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.
ServerHostName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the node name that corresponds to the application server instance being monitored, and specify that name against the ServerHostName parameter.
NDManager	The NDManager parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host

Parameters	Description
	In the case of situation (a), in the NDManager text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the Node Manager, do the following: • Login to the Administrative Console of the node manager using the URL http://<IPoftheWebSphereAppServer'sNodeManager>:<Adminport>/admin/. • Using the tree-structure in the left pane of the Administrative Console that appears, drill down to the **Deployment Manager** node within **System Administration**. • Select the **Configuration** tab that appears in the right pane, and scroll down to the **End Points** link in the **Additional Properties** section. • Once you locate the **End Points** link, click on it, and in the page that appears, click on the **SOAP_CONNECTOR_ADDRESS** link. • In the subsequent page, a fully qualified domain name will be displayed against Host. This is the name that should be specified as the host name of the node manager in the NDManager text box. In the case of situation (b), enter the ServerHostName itself as the NDManager. If both conditions (a) and (b) do not apply, then specify *none* here.
ConnectorPort	The ConnectorPort parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In case of situation (a), the ConnectorPort parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the following: ◦ From the Node manager's host, open the <WEBSPHERE_ INSTALL_ DIR>\DeploymentManager\properties\wsadmin.properties file. ◦ Two entries that read as follows will be present in the wsadmin.properties file: *com.ibm.ws.scripting.port=<port_number>* *#com.ibm.ws.scripting.port=<port_number>* Both the entries will have port numbers against them. However, the

Parameters	Description
	uncommented entry (the entry without the #) is the one that denotes an active port. Therefore, specify the corresponding port number as the ConnectorPort. In case of situation (b), do the following to know the ConnectorPort: • Login to the WebSphere Administrative Console. • Expand the Servers node in the tree-structure in the left pane of the console, and click on the Application Servers link. A list of application server instances will then appear in the right pane. • Click on the server instance being monitored. Doing so invokes the **Configuration** of the chosen server instance appears. • Scroll down the **Configuration** to view the **End Points** link. Once you locate the **End Points** link, click on it. • In the page that appears next, click on the **SOAP_CONNECTOR_ADDRESS** link. • The Port displayed in that page should be used as the ConnectorPort in situation (b). If both (a) and (b) do not apply, specify *none* against ConnectorPort.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid User name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the User text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.
EncryptPass	By default, this flag is set to **Yes**, indicating that the Password of the WebSphere server is encrypted by default. To disable password encryption, select the **No** option.
ServerName	Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, first, connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere**

Parameters	Description
	server:Port number of the WebSphere server>\admin. Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node. A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane. One of the displayed server instances can be specified as the value of the ServerName parameter.
Timeout	In the Timeout text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default Timeout period is 60 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Local transaction begin rate	Indicates the rate at which the local transactions begin at the server.	Transactions/Sec	A high value indicates an increased overload on the server.
Active local transactions	Indicates the average number of concurrently active local transactions.	Number	If the value of this measure is high, it signifies greater load on the server. This might increase the number of waiting local transactions.
Local transaction duration	Indicates the average duration of the local transactions.	Secs	A high transaction duration value indicates increased load on the server.
Local commit duration	Indicates the average duration of commit for local transactions.	Secs	If the total time taken for committing a local transaction is high, then it indicates a bottleneck on the server.
Local transaction commits	Indicates the rate at which the local transactions are being committed.	Trans/Sec	If the rate of transactions that are being committed is very high, it signifies load on the server. It might be caused when some locked transactions are released suddenly.
Local transaction rollbacks	Indicates the rate at which the local transactions are being rolled back.	Trans/Sec	A high value indicates a problem with the application. Possible reasons could be the problem with application or with

Measurement	Description	Measurement Unit	Interpretation
			some other dependent server (e.g. Database).
Local transaction timeouts:	Indicates the rate at which the local transactions are being timed out.	Trans/Sec	Again, a rise in the value could be due to the problem with the application or with some other dependent server like the database.

4.4.3 WebSphere Servlet Sessions Test

Http sessions form a part of any business application. This test monitors the http sessions on the WebSphere application server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each WebSphere application server being monitored

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WebServerPort	The port number through which the WebSphere applications can be accessed.
SSL	Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.
ServerHostName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then

Parameters	Description
	appear in the right pane of the console. From this list, you can figure out the node name that corresponds to the application server instance being monitored, and specify that name against the ServerHostName parameter.
NDManager	The NDManager parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In the case of situation (a), in the NDManager text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the Node Manager, do the following: • Login to the Administrative Console of the node manager using the URL http://<IPoftheWebSphereAppServer'sNodeManager>:<Adminport>/admin/. • Using the tree-structure in the left pane of the Administrative Console that appears, drill down to the **Deployment Manager** node within **System Administration**. • Select the **Configuration** tab that appears in the right pane, and scroll down to the **End Points** link in the **Additional Properties** section. • Once you locate the **End Points** link, click on it, and in the page that appears, click on the **SOAP_CONNECTOR_ADDRESS** link. • In the subsequent page, a fully qualified domain name will be displayed against Host. This is the name that should be specified as the host name of the node manager in the NDManager text box. In the case of situation (b), enter the ServerHostName itself as the NDManager. If both conditions (a) and (b) do not apply, then specify *none* here.
ConnectorPort	The ConnectorPort parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In case of situation (a), the ConnectorPort parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the

Parameters	Description
	following: ◦ From the Node manager's host, open the <WEBSPHERE_ INSTALL_ DIR>\DeploymentManager\properties\wsadmin.properties file. ◦ Two entries that read as follows will be present in the wsadmin.properties file: *com.ibm.ws.scripting.port=<port_number>* *#com.ibm.ws.scripting.port=<port_number>* Both the entries will have port numbers against them. However, the uncommented entry (the entry without the #) is the one that denotes an active port. Therefore, specify the corresponding port number as the ConnectorPort. In case of situation (b), do the following to know the ConnectorPort: • Login to the WebSphere Administrative Console. • Expand the Servers node in the tree-structure in the left pane of the console, and click on the Application Servers link. A list of application server instances will then appear in the right pane. • Click on the server instance being monitored. Doing so invokes the **Configuration** of the chosen server instance appears. • Scroll down the **Configuration** to view the **End Points** link. Once you locate the **End Points** link, click on it. • In the page that appears next, click on the **SOAP_CONNECTOR_ADDRESS** link. • The Port displayed in that page should be used as the ConnectorPort in situation (b). If both (a) and (b) do not apply, specify *none* against ConnectorPort.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid User name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the User text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.

Parameters	Description
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.
EncryptPass	By default, this flag is set to **Yes**, indicating that the Password of the WebSphere server is encrypted by default. To disable password encryption, select the **No** option.
ServerName	Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, first, connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node. A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane. One of the displayed server instances can be specified as the value of the ServerName parameter.
Timeout	In the Timeout text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default Timeout period is 60 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Session creation rate	Indicates the rate at which the sessions are created in the server.	Sessions/Sec	A high value indicates a high level of activity in this application. By observing the variations to this measure over time, you can determine usage trends such as the time of day when an application is getting the most amount of traffic.
Invalidated sessions	Indicates the number of sessions that were invalidated.	Number	This message is indicative of the session usage pattern of this application.
Session life time	This measure indicates the average lifetime of a session.	Secs	A long session lifetime may occur due to one of the following reasons: 1. The application may not timeout the sessions after a fixed interval of time.

Measurement	Description	Measurement Unit	Interpretation
			2. The sessions in the application might be active for a very long time.
Active sessions	This measure indicates the total number of sessions that currently are being accessed by the requests.	Number	A high value indicates an increased workload on the server.
Live sessions	Indicates the total number of valid sessions on the server.	Number	A high value indicates an increased workload on the server.

4.4.4 WebSphere Web Applications Test

A web application is a collection of different web components such as servlets, JSPs etc. This test tracks the statistics pertaining to the web applications hosted on the WebSphere application server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each web application that is deployed in the WebSphere application server being monitored

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WebServerPort	The port number through which the WebSphere applications can be accessed.
SSL	Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.
ServerHostName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console.

Parameters	Description
	• Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the node name that corresponds to the application server instance being monitored, and specify that name against the ServerHostName parameter.
NDManager	The NDManager parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In the case of situation (a), in the NDManager text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the Node Manager, do the following: • Login to the Administrative Console of the node manager using the URL http://<IPoftheWebSphereAppServer'sNodeManager>:<Adminport>/admin/. • Using the tree-structure in the left pane of the Administrative Console that appears, drill down to the **Deployment Manager** node within **System Administration**. • Select the **Configuration** tab that appears in the right pane, and scroll down to the **End Points** link in the **Additional Properties** section. • Once you locate the **End Points** link, click on it, and in the page that appears, click on the **SOAP_CONNECTOR_ADDRESS** link. • In the subsequent page, a fully qualified domain name will be displayed against Host. This is the name that should be specified as the host name of the node manager in the NDManager text box. In the case of situation (b), enter the ServerHostName itself as the NDManager. If both conditions (a) and (b) do not apply, then specify *none* here.
ConnectorPort	The ConnectorPort parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running

Parameters	Description
	on the same host In case of situation (a), the ConnectorPort parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the following: ◦ From the Node manager's host, open the <WEBSPHERE_ INSTALL_ DIR>\DeploymentManager\properties\wsadmin.properties file. ◦ Two entries that read as follows will be present in the wsadmin.properties file: *com.ibm.ws.scripting.port=<port_number>* *#com.ibm.ws.scripting.port=<port_number>* Both the entries will have port numbers against them. However, the uncommented entry (the entry without the #) is the one that denotes an active port. Therefore, specify the corresponding port number as the ConnectorPort. In case of situation (b), do the following to know the ConnectorPort: • Login to the WebSphere Administrative Console. • Expand the Servers node in the tree-structure in the left pane of the console, and click on the Application Servers link. A list of application server instances will then appear in the right pane. • Click on the server instance being monitored. Doing so invokes the **Configuration** of the chosen server instance appears. • Scroll down the **Configuration** to view the **End Points** link. Once you locate the **End Points** link, click on it. • In the page that appears next, click on the **SOAP_CONNECTOR_ADDRESS** link. • The Port displayed in that page should be used as the ConnectorPort in situation (b). If both (a) and (b) do not apply, specify *none* against ConnectorPort.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid User name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the User text box should contain the default value 'none'.

Parameters	Description
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.
EncryptPass	By default, this flag is set to **Yes**, indicating that the Password of the WebSphere server is encrypted by default. To disable password encryption, select the **No** option.
ServerName	Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, first, connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node. A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane. One of the displayed server instances can be specified as the value of the ServerName parameter.
Timeout	In the Timeout text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default Timeout period is 60 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Loaded servlets	Indicates the current number of loaded servlets.	Number	A sudden increase in the number of loaded servlets indicates an increase in the load on the server or for that particular application.
Reloads	Indicates the number of times the servlet is being reloaded.	Number	It is preferable to have minimum reloads for optimal performance of the application.
Requests	Indicates the total number of requests for the servlets.	Reqs/Sec	An increase in the request rate indicates an increase in the server load.
Concurrent requests	Number of requests that	Number	An increase in the value of this

Measurement	Description	Measurement Unit	Interpretation
	are concurrently being processed.		measure indicates an increase in the user workload. It may also increase due to the overload on the server.
Avg response time	Indicates the time taken for the completion of the requests.	Secs	An increasing value indicates a bottleneck on the server. This may arise due to one of the following reasons: • The server might be running out of resources. • The server may be overloaded with requests. • Some problem with the available bandwidth on the network.
Errors	Indicates the rate at which errors or exceptions are occurring in the servlets.	Errors/Sec	A high rate of errors might occur due to too many requests on the application or there might be a bottleneck on the application.

4.4.5 WebSphere ORB Summary Test

This test reports statistics pertaining to the ORBs (Object Request Broker) on a WebSphere server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each server instance being monitored

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed .
Host	The IP address of the WebSphere application server

Parameters	Description
Port	The port number of the WebSphere application server
WebServerPort	The port number through which the WebSphere applications can be accessed.
SSL	Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.
ServerHostName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the node name that corresponds to the application server instance being monitored, and specify that name against the ServerHostName parameter.
NDManager	The NDManager parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In the case of situation (a), in the NDManager text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the Node Manager, do the following: • Login to the Administrative Console of the node manager using the URL http://<IPoftheWebSphereAppServer'sNodeManager>:<Adminport>/admin/. • Using the tree-structure in the left pane of the Administrative Console that appears, drill down to the **Deployment Manager** node within **System Administration**. • Select the **Configuration** tab that appears in the right pane, and scroll down to the **End Points** link in the **Additional Properties** section. • Once you locate the **End Points** link, click on it, and in the page that appears, click on the **SOAP_CONNECTOR_ADDRESS** link. • In the subsequent page, a fully qualified domain name will be displayed

Parameters	Description
	against Host. This is the name that should be specified as the host name of the node manager in the NDManager text box. In the case of situation (b), enter the ServerHostName itself as the NDManager. If both conditions (a) and (b) do not apply, then specify *none* here.
ConnectorPort	The ConnectorPort parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In case of situation (a), the ConnectorPort parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the following: ◦ From the Node manager's host, open the <WEBSPHERE_ INSTALL_ DIR>\DeploymentManager\properties\wsadmin.properties file. ◦ Two entries that read as follows will be present in the wsadmin.properties file: *com.ibm.ws.scripting.port=<port_number>* *#com.ibm.ws.scripting.port=<port_number>* Both the entries will have port numbers against them. However, the uncommented entry (the entry without the #) is the one that denotes an active port. Therefore, specify the corresponding port number as the ConnectorPort. In case of situation (b), do the following to know the ConnectorPort: • Login to the WebSphere Administrative Console. • Expand the Servers node in the tree-structure in the left pane of the console, and click on the Application Servers link. A list of application server instances will then appear in the right pane. • Click on the server instance being monitored. Doing so invokes the **Configuration** of the chosen server instance appears. • Scroll down the **Configuration** to view the **End Points** link. Once you locate the **End Points** link, click on it.

Parameters	Description
	• In the page that appears next, click on the **SOAP_CONNECTOR_ADDRESS** link. • The Port displayed in that page should be used as the ConnectorPort in situation (b). If both (a) and (b) do not apply, specify *none* against ConnectorPort.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid User name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the User text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.
EncryptPass	By default, this flag is set to **Yes**, indicating that the Password of the WebSphere server is encrypted by default. To disable password encryption, select the **No** option.
ServerName	Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, first, connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node. A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane. One of the displayed server instances can be specified as the value of the ServerName parameter.
Timeout	In the Timeout text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default Timeout period is 60 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Reference lookup time:	The amount of time taken to lookup an object reference before method dispatch can be carried out.	Secs	An excessively long time may indicate an EJB container lookup problem.
Total requests:	Indicates the total number of requests sent to the ORB.	Number	
Concurrent requests:	Indicates the number of requests that are concurrently processed by the ORB.	Number	
Avg processing time:	Indicates the time it takes a registered portable interceptor to run.	Secs	

4.4.6 WebSphere Web Applications Summary Test

This test reports statistics pertaining to the web applications deployed on a WebSphere server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each server instance being monitored

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WebServerPort	The port number through which the WebSphere applications can be accessed.
SSL	Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.
ServerHostName	Specify the node name of the server instance being monitored. To know the node name,

Parameters	Description
	do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the node name that corresponds to the application server instance being monitored, and specify that name against the ServerHostName parameter.
NDManager	The NDManager parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In the case of situation (a), in the NDManager text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the Node Manager, do the following: • Login to the Administrative Console of the node manager using the URL http://<IPoftheWebSphereAppServer'sNodeManager>:<Adminport>/admin/. • Using the tree-structure in the left pane of the Administrative Console that appears, drill down to the **Deployment Manager** node within **System Administration**. • Select the **Configuration** tab that appears in the right pane, and scroll down to the **End Points** link in the **Additional Properties** section. • Once you locate the **End Points** link, click on it, and in the page that appears, click on the **SOAP_CONNECTOR_ADDRESS** link. • In the subsequent page, a fully qualified domain name will be displayed against Host. This is the name that should be specified as the host name of the node manager in the NDManager text box. In the case of situation (b), enter the ServerHostName itself as the NDManager. If both conditions (a) and (b) do not apply, then specify *none* here.
ConnectorPort	The ConnectorPort parameter is applicable only under the following circumstances:

Parameters	Description
	a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In case of situation (a), the ConnectorPort parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the following: ◦ From the Node manager's host, open the <WEBSPHERE_INSTALL_DIR>\DeploymentManager\properties\wsadmin.properties file. ◦ Two entries that read as follows will be present in the wsadmin.properties file: *com.ibm.ws.scripting.port=<port_number>* *#com.ibm.ws.scripting.port=<port_number>* Both the entries will have port numbers against them. However, the uncommented entry (the entry without the #) is the one that denotes an active port. Therefore, specify the corresponding port number as the ConnectorPort. In case of situation (b), do the following to know the ConnectorPort: • Login to the WebSphere Administrative Console. • Expand the Servers node in the tree-structure in the left pane of the console, and click on the Application Servers link. A list of application server instances will then appear in the right pane. • Click on the server instance being monitored. Doing so invokes the **Configuration** of the chosen server instance appears. • Scroll down the **Configuration** to view the **End Points** link. Once you locate the **End Points** link, click on it. • In the page that appears next, click on the **SOAP_CONNECTOR_ADDRESS** link. • The Port displayed in that page should be used as the ConnectorPort in situation (b). If both (a) and (b) do not apply, specify *none* against ConnectorPort.
User	If security has been enabled for the WebSphere server being monitored, then provide a

Parameters	Description
	valid User name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the User text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.
EncryptPass	By default, this flag is set to **Yes**, indicating that the Password of the WebSphere server is encrypted by default. To disable password encryption, select the **No** option.
ServerName	Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, first, connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node. A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane. One of the displayed server instances can be specified as the value of the ServerName parameter.
Timeout	In the Timeout text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default Timeout period is 60 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Loaded servlets	Indicates the number of servlets that were loaded.	Number	
Reloads	Indicates the number of servlets that were reloaded.	Number	
Total requests	Indicates the number of requests that the servlets have processed.	Number	

Measurement	Description	Measurement Unit	Interpretation
Concurrent requests	Indicates the number of requests that the servlets have concurrently processed.	Number	
Avg response time	Indicates the time it takes for a servlet request to be serviced.	Secs	
Total errors	Indicates the total number of errors in the servlet/JSP.	Number	

4.4.7 WebSphere Web Server Summary Test

This test reports statistics pertaining to the web services mounted on a WebSphere server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each server instance being monitored

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WebServerPort	The port number through which the WebSphere applications can be accessed.
SSL	Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.
ServerHostName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click

Parameters	Description
	on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the node name that corresponds to the application server instance being monitored, and specify that name against the ServerHostName parameter.
NDManager	The NDManager parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In the case of situation (a), in the NDManager text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the Node Manager, do the following: • Login to the Administrative Console of the node manager using the URL http://<IPoftheWebSphereAppServer'sNodeManager>:<Adminport>/admin/. • Using the tree-structure in the left pane of the Administrative Console that appears, drill down to the **Deployment Manager** node within **System Administration**. • Select the **Configuration** tab that appears in the right pane, and scroll down to the **End Points** link in the **Additional Properties** section. • Once you locate the **End Points** link, click on it, and in the page that appears, click on the **SOAP_CONNECTOR_ADDRESS** link. • In the subsequent page, a fully qualified domain name will be displayed against Host. This is the name that should be specified as the host name of the node manager in the NDManager text box. In the case of situation (b), enter the ServerHostName itself as the NDManager. If both conditions (a) and (b) do not apply, then specify *none* here.
ConnectorPort	The ConnectorPort parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host

Parameters	Description
	In case of situation (a), the ConnectorPort parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the following: ◦ From the Node manager's host, open the <WEBSPHERE_INSTALL_DIR>\DeploymentManager\properties\wsadmin.properties file. ◦ Two entries that read as follows will be present in the wsadmin.properties file: *com.ibm.ws.scripting.port=<port_number>* *#com.ibm.ws.scripting.port=<port_number>* Both the entries will have port numbers against them. However, the uncommented entry (the entry without the #) is the one that denotes an active port. Therefore, specify the corresponding port number as the ConnectorPort. In case of situation (b), do the following to know the ConnectorPort: • Login to the WebSphere Administrative Console. • Expand the Servers node in the tree-structure in the left pane of the console, and click on the Application Servers link. A list of application server instances will then appear in the right pane. • Click on the server instance being monitored. Doing so invokes the **Configuration** of the chosen server instance appears. • Scroll down the **Configuration** to view the **End Points** link. Once you locate the **End Points** link, click on it. • In the page that appears next, click on the **SOAP_CONNECTOR_ADDRESS** link. • The Port displayed in that page should be used as the ConnectorPort in situation (b). If both (a) and (b) do not apply, specify *none* against ConnectorPort.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid User name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the User text box should contain the default value 'none'.

Parameters	Description
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.
EncryptPass	By default, this flag is set to **Yes**, indicating that the Password of the WebSphere server is encrypted by default. To disable password encryption, select the **No** option.
ServerName	Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, first, connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node. A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane. One of the displayed server instances can be specified as the value of the ServerName parameter.
Timeout	In the Timeout text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default Timeout period is 60 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Loaded services	Indicates the number of web services loaded by the application server.	Number	
Requests received	Indicates the number of requests received by the web services.	Number	
Requests dispatched	Indicates the number of requests dispatched by the service to a target code.	Number	
Requests successful	Indicates the number of requests that were	Number	

Measurement	Description	Measurement Unit	Interpretation
	successfully responded to.		
Avg response time	Indicates the average time between receipt of a request and the dispatch of a response.	Secs	

4.4.8 WebSphere ORB Test

This test reports statistics pertaining to each of the ORBs (Object Request Broker) on a WebSphere server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each server instance being monitored

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WebServerPort	The port number through which the WebSphere applications can be accessed.
SSL	Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.
ServerHostName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the node

Parameters	Description
	name that corresponds to the application server instance being monitored, and specify that name against the ServerHostName parameter.
NDManager	The NDManager parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In the case of situation (a), in the NDManager text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the Node Manager, do the following: • Login to the Administrative Console of the node manager using the URL http://<IPoftheWebSphereAppServer'sNodeManager>:<Adminport>/admin/. • Using the tree-structure in the left pane of the Administrative Console that appears, drill down to the **Deployment Manager** node within **System Administration**. • Select the **Configuration** tab that appears in the right pane, and scroll down to the **End Points** link in the **Additional Properties** section. • Once you locate the **End Points** link, click on it, and in the page that appears, click on the **SOAP_CONNECTOR_ADDRESS** link. • In the subsequent page, a fully qualified domain name will be displayed against Host. This is the name that should be specified as the host name of the node manager in the NDManager text box. In the case of situation (b), enter the ServerHostName itself as the NDManager. If both conditions (a) and (b) do not apply, then specify *none* here.
ConnectorPort	The ConnectorPort parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In case of situation (a), the ConnectorPort parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the following:

Parameters	Description
	◦ From the Node manager's host, open the <WEBSPHERE_INSTALL_DIR>\DeploymentManager\properties\wsadmin.properties file. ◦ Two entries that read as follows will be present in the wsadmin.properties file: *com.ibm.ws.scripting.port=<port_number>* *#com.ibm.ws.scripting.port=<port_number>* Both the entries will have port numbers against them. However, the uncommented entry (the entry without the #) is the one that denotes an active port. Therefore, specify the corresponding port number as the ConnectorPort. In case of situation (b), do the following to know the ConnectorPort: • Login to the WebSphere Administrative Console. • Expand the Servers node in the tree-structure in the left pane of the console, and click on the Application Servers link. A list of application server instances will then appear in the right pane. • Click on the server instance being monitored. Doing so invokes the **Configuration** of the chosen server instance appears. • Scroll down the **Configuration** to view the **End Points** link. Once you locate the **End Points** link, click on it. • In the page that appears next, click on the **SOAP_CONNECTOR_ADDRESS** link. • The Port displayed in that page should be used as the ConnectorPort in situation (b). If both (a) and (b) do not apply, specify *none* against ConnectorPort.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid User name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the User text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the

Parameters	Description
	Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.
EncryptPass	By default, this flag is set to **Yes**, indicating that the Password of the WebSphere server is encrypted by default. To disable password encryption, select the **No** option.
ServerName	Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, first, connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node. A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane. One of the displayed server instances can be specified as the value of the ServerName parameter.
Timeout	In the Timeout text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default Timeout period is 60 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Avg processing time	Indicates the time this registered portable interceptor takes to run.	Secs	A high processing time is indicative of a performance bottleneck.

4.4.9 WebSphere Web Server Test

This test reports statistics pertaining to each of the web services mounted on a WebSphere server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each web service on the monitored WebSphere server instance

Configurable parameters for the test

Parameters	Description
Test Period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WebServerPort	The port number through which the WebSphere applications can be accessed.
SSL	Indicate whether the SSL (Secured Socket Layer) is to be used to connect to the WebSphere server.
ServerHostName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the node name that corresponds to the application server instance being monitored, and specify that name against the ServerHostName parameter.
NDManager	The NDManager parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In the case of situation (a), in the NDManager text box, provide the host name of the node manager that manages the application servers in the cluster. To know the name of the Node Manager, do the following: • Login to the Administrative Console of the node manager using the URL http://<IPoftheWebSphereAppServer'sNodeManager>:<Adminport>/admin/. • Using the tree-structure in the left pane of the Administrative Console that appears, drill down to the **Deployment Manager** node within **System Administration**. • Select the **Configuration** tab that appears in the right pane, and scroll down to

Parameters	Description
	the **End Points** link in the **Additional Properties** section. • Once you locate the **End Points** link, click on it, and in the page that appears, click on the **SOAP_CONNECTOR_ADDRESS** link. • In the subsequent page, a fully qualified domain name will be displayed against Host. This is the name that should be specified as the host name of the node manager in the NDManager text box. In the case of situation (b), enter the ServerHostName itself as the NDManager. If both conditions (a) and (b) do not apply, then specify *none* here.
ConnectorPort	The ConnectorPort parameter is applicable only under the following circumstances: a. If the WebSphere server being monitored belongs to a cluster, or, b. If the WebSphere server being monitored is one of many instances of the server running on the same host In case of situation (a), the ConnectorPort parameter will take the port number using which the node manager communicates with the WebSphere servers in the cluster. The connector port can be a SOAP port or an RMI port. To know the port number, do the following: ◦ From the Node manager's host, open the <WEBSPHERE_INSTALL_DIR>\DeploymentManager\properties\wsadmin.properties file. ◦ Two entries that read as follows will be present in the wsadmin.properties file: *com.ibm.ws.scripting.port=<port_number>* *#com.ibm.ws.scripting.port=<port_number>* Both the entries will have port numbers against them. However, the uncommented entry (the entry without the #) is the one that denotes an active port. Therefore, specify the corresponding port number as the ConnectorPort. In case of situation (b), do the following to know the ConnectorPort: • Login to the WebSphere Administrative Console. • Expand the Servers node in the tree-structure in the left pane of the console, and click on the Application Servers link. A list of application server instances will then appear in the right pane. • Click on the server instance being monitored. Doing so invokes the **Configuration** of

Parameters	Description
	the chosen server instance appears. • Scroll down the **Configuration** to view the **End Points** link. Once you locate the **End Points** link, click on it. • In the page that appears next, click on the **SOAP_CONNECTOR_ADDRESS** link. • The Port displayed in that page should be used as the ConnectorPort in situation (b). If both (a) and (b) do not apply, specify *none* against ConnectorPort.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid User name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the User text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.
EncryptPass	By default, this flag is set to **Yes**, indicating that the Password of the WebSphere server is encrypted by default. To disable password encryption, select the **No** option.
ServerName	Specify the name of the WebSphere server instance to be monitored. To know the instances of a WebSphere server currently available, first, connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. Then, login to the administrative console and expand the **Servers** node in the left pane of the console. Next, click on the **Application Servers** sub-node under the **Servers** node. A list of server instance **Names** and their corresponding **Node** values will then be displayed in the right pane. One of the displayed server instances can be specified as the value of the ServerName parameter.
Timeout	In the Timeout text box, specify the maximum duration (in seconds) for which the test will wait for a response from the application server. The default Timeout period is 60 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Loaded services	Indicates the number of instances of this web service that have been loaded by the application server.	Number	
Requests received	Indicates the number of requests received by this web service.	Number	
Requests dispatched	Indicates the number of requests dispatched by this web service to a target code.	Number	
Requests successful	Indicates the number of requests that were succesfully responded to by this web service.	Number	
Avg response time	Indicates the average time between receipt of a request and the dispatch of a response by this web service.	Secs	

4.5 The Java Transactions Layer

By default, this layer will not be available for the IBM WebSphere application server. This is because, the **Java Business Transactions** test mapped to this layer is disabled by default. To enable the test, follow the *Agents -> Tests -> Enable/Disable* menu sequence, select *IBM WebSphere* as the **Component type**, *Performance* as the **Test type**, and then select **Java Business Transactions** from the **DISABLED TESTS** list. Click the **Enable** button to enable the selected test, and click the **Update** button to save the changes.

4.5.1 Java Business Transactions Test

The responsiveness of a transaction is the key determinant of user experience with that transaction; if response time increases, user experience deteriorates. To make users happy, a Java business transaction should be rapidly processed by each of the JVM nodes in its path. Processing bottlenecks on a single JVM node can slowdown/stall an entire business transaction or can cause serious transaction errors. This in turn can badly scar the experience of users. To avoid this,

administrators should promptly identify slow/stalled/errored transactions, isolate the JVM node on which the slowness/error occurred, and uncover what caused the aberration on that node – is it owing to SQL queries executed by the node? Or is it because of external calls – eg., async calls, SAP JCO calls, HTTP calls, etc. - made by that node? The **Java Business Transactions** test helps with this!

This test runs on a BTM-enabled JVM in an IT infrastructure, tracks all the transaction requests received by that JVM, and groups requests based on user-configured pattern specifications. For each transaction pattern, the test then computes and reports the average time taken by that JVM node to respond to the transaction requests of that pattern. In the process, the test identifies the slow/stalled transactions of that pattern, and reports the count of such transactions and their responsiveness. Detailed diagnostics provided by the test accurately pinpoint the exact transaction URLs that are slow/stalled, the total round-trip time of each transaction, and also indicate when such transaction requests were received by that node. The slowest transaction in the group can thus be identified.

For this test to run and report metrics on a WebSphere server, you first need to BTM-enable the WebSphere JVM. To know how, refer to the Installing eG Java BTM on an IBM WebSphere topic in the *Java Business Transaction Monitoring* document.

Then, proceed to configure this test. Refer to the Java Business Transactions Test topic in the *Java Business Transaction Monitoring* document to know how to configure this test and learn about the metrics it reports.

Chapter 5: Troubleshooting

If any of the WebSphere tests is not reporting measures, then try to connect to the URL of the following format:

http://<WebSphereIP>:<WebSpherePort>/ egurkha/egurkha/EgWebSphere.jsp?module=<moduleName> &summary=true&hostname=<nodeName>&server=<servername>&user=<userName>&password=<password>&ndmanager=<nodemanagerhost>&connectorport=<connectorport>

Here, specify

hostIP = IP of the machine in which the WebSphere Server is running.

hostPort = The port on which the server is running.

moduleName = The name of the module for which measures are required. This will vary from one test to another.

nodeName= The name of the node for every server

servername = The value of the **SERVERNAME** parameter of the test

user and password = If security is enabled for the WebSphere server being monitored, then provide a valid user name and password

ndmanager and connectorport = If the WebSphere server being monitored belongs to a WebSphere cluster, then provide the values of the ndmanager and connectorport parameters.

Examples:

Note:

All the examples provided below pertain to non-clustered WebSphere environments, where security is not enabled.

1. If the WebSphere JDBC test is not reporting measures, then use the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=connectionPoolModule&summary=true&hostname=egitlab04&server=server1

2. If the WebSphereGlobalTransactions test is not reporting measures, then connect to the

following URL:

http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=transactionModule&summary=true&hostname=egitlab04&transaction=global&server= server1

3. If the WebSphere Local Transactions test is not reporting measures, then connect to the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=transactionModule&summary=true&hostname=egitlab04&transaction=local&server= server1

4. If the WebSphere Servlet Sessions test is not reporting measures, then connect to the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=servletSessionsModule&summary=true&hostname=egitlab04&server= server1

5. If the WebSphere Web Applications test is not reporting measures, then connect to the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=webAppModule&summary=true&hostname=egitlab04&webapps=true&servlets=false&server= server1

6. If the WebSphere Thread Pools test is not reporting measures, then connect to the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=threadPoolModule&summary=true&hostname=egitlab04&server= server1

7. If the WebSphere JVM test is not reporting measures, then connect to the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=jvmRuntimeModule&summary=true&hostname=egitlab04&server= server1

8. If the Ws Beans test is not reporting measures, then connect to the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=OraclenModule&summary=true&hostname=egitlab04&server= server1

9. If the WebSphere ORB Summary test is not reporting measures, then connect to the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=orbPerfModule&summary=false&hostname=egitlab04&server= server1

10. If the WebSphere ORB test is not reporting measures, then connect to the following URL:

    http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=orbPerfModule&summary=false&hostname=egitlab04&server= server1

11. If the WebSphere Web Applications Summary test is not reporting measures, then connect to the following URL:

    http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=webAppModule&summary=true&hostname=egitlab04&webapps=true&servlets=false&server= server1

12. If the WsWebSrvSum test is not reporting measures, then connect to the following URL:

    http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=webAppModule&summary=true&hostname=egitlab04&webapps=true&servlets=falseserver= server1

13. If the WsWebSrv test is not reporting measures, then connect to the following URL:

    http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=webAppModule&summary=true&hostname=egitlab04&webapps=true&servlets=falseserver= server1

14. If the WebSphere test is not reporting measures, then connect to the following URL:

    http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=jvmRuntimeModule&summary=true&hostname=egitlab04&server= server1

If the desired output is not available in the above-mentioned URLs, then you can troubleshoot further using the URL:

http://<WebSphereIP>:<WebSpherePort>/<moduleName>&summary=true&hostname=<nodeName>&server=<servername>&debug=true

For example, if the URL for WebSphere test does not display the performance metrics, then use the following URL to look for corresponding error information:

http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere.jsp?module=jvmRuntimeModule&summary=true&hostname=egitlab04&server= server1&debug=true

If these URLs do not display the required statistics, then it could indicate either/all of the following:

- The ear file has not been deployed properly. To verify this, follow the procedure discussed in Section **2.7**.
- The WebSphere server has not been started. In which case, you will have to start the WebSphere application server.

Chapter 6: Monitoring the WebSphere Application Server 6.0 (and above)

As the WebSphere server 6.0 (and above) supports different capabilities and interfaces than the earlier versions, the eG Enterprise suite offers a different monitoring model for WebSphere 6.0 (and above). In the case of the WebSphere application server version 6.0 and above, the eG agent uses the JMX architecture supported by the WebSphere server to gather the required metrics.

To monitor a WebSphere Application server 6.0 (and above), a specific WebSphere monitoring eG component has to be installed on it. If the WebSphere server to be monitored uses a SOAP connector port for its internal communications, then the **egurkha6.ear** application (in the **/opt/egurkha/lib** or the **<EG_INSTALL_DIR>\lib** directory) will have to be installed on the server. On the other hand, if the target WebSphere server uses an RMI port for communicating internally, then the **egurkha6rmi.ear** application should be installed on the server. The procedure for installing the 'ear' file has been discussed in detail in the *eG Implementer's Guide*.

The measures extracted by **egurkha6.ear** or **egurkha6rmi.ear** (as the case may be) are captured and displayed by the unique monitoring model that eG Enterprise prescribes for the WebSphere Appolication server 6.0 (and above) (see Figure 6.1).

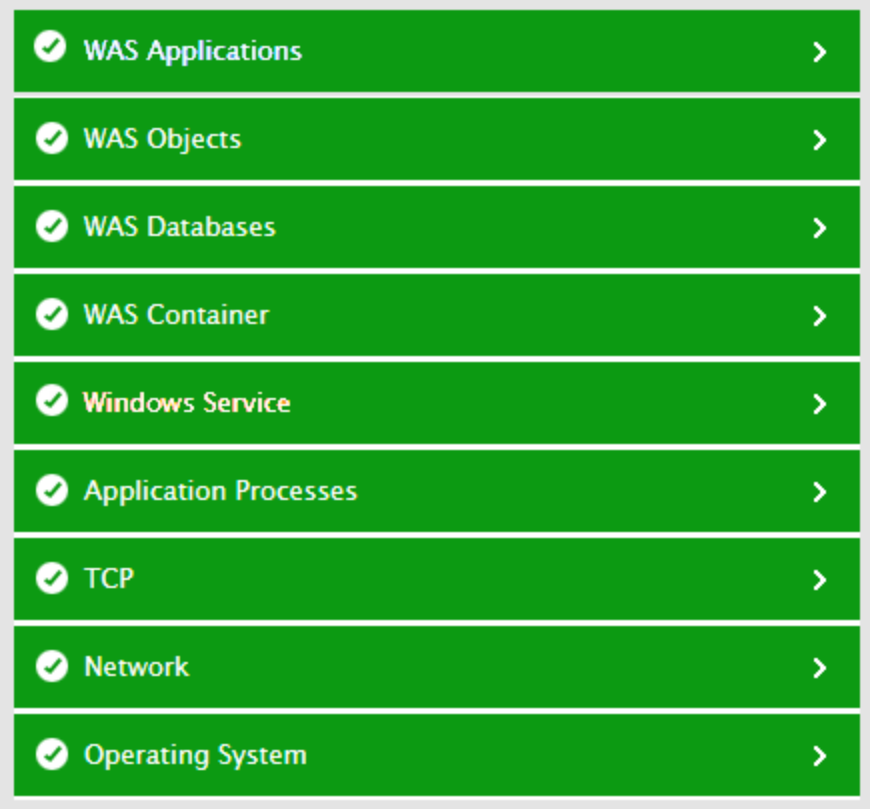

Figure 6.1: Layer model of the WebSphere Application server 6.0 (or above)

The tests executing on each of the layers depicted by Figure 6.1 extract a wealth of performance data from the WebSphere application server, that enable WebSphere administrators to determine the following:

- Does EJB creation take too long?
- Was the WebSphere cache adequately sized, or were there too many cache misses?
- Is sufficient memory allotted to the WebSphere JVM?
- Is the thread pool adequately sized?
- Did too many connections in the database connection pool timeout?
- Are transaction rollbacks happening too frequently on the server?
- Are transactions taking too long to complete?
- How many synchronous/asynchronous requests were processed by the WebSphere gateway?
- How well does the ORB (Object Request Broker) handle requests for objects received by it?
- Were any errors detected during JSP/servlet processing?

- Are the web services executing on the WebSphere server too slow in responding to requests received by it?

The sections to come will elaborate on the tests associated with the top 5 layers only.

6.1 The JVM Layer

By default, no tests are mapped to the **JVM** layer of the WebSphere Application server 6.0 (or above). This is because, all tests reporting critical JVM-related metrics are disabled by default for the WebSphere server. To enable one/more of these tests, open the **AGENTS – TESTS CONFIGURATION** page using the Agents -> Tests -> Configure menu sequence, select **WebSphere Application** from the **Component type** list, scroll down the test list that appears to view the **DISABLED TESTS** section, select the check box corresponding to the JVM test of interest to you, and then, click the **Update** button.

If all the JVM tests are enabled, then clicking on the **JVM** layer will display the test list depicted by Figure 6.2 below.

Figure 6.2: The tests mapped to the JVM layer

These tests collectively report the resource usage and overall health of the WebSphere JVM. These tests have been elaborately discussed in Monitoring Java Applications document. The forthcoming sections deal with the tests pertaining to the WebSphere Application Server.

6.2 The WAS Container Layer

Using the tests associated with the **WAS Container** layer, the eG agents can closely observe the performance of the following critical services executing on the *WebSphere Application* server:

- Cache management
- Object pools
- Garbage collection (GC)
- Thread pools

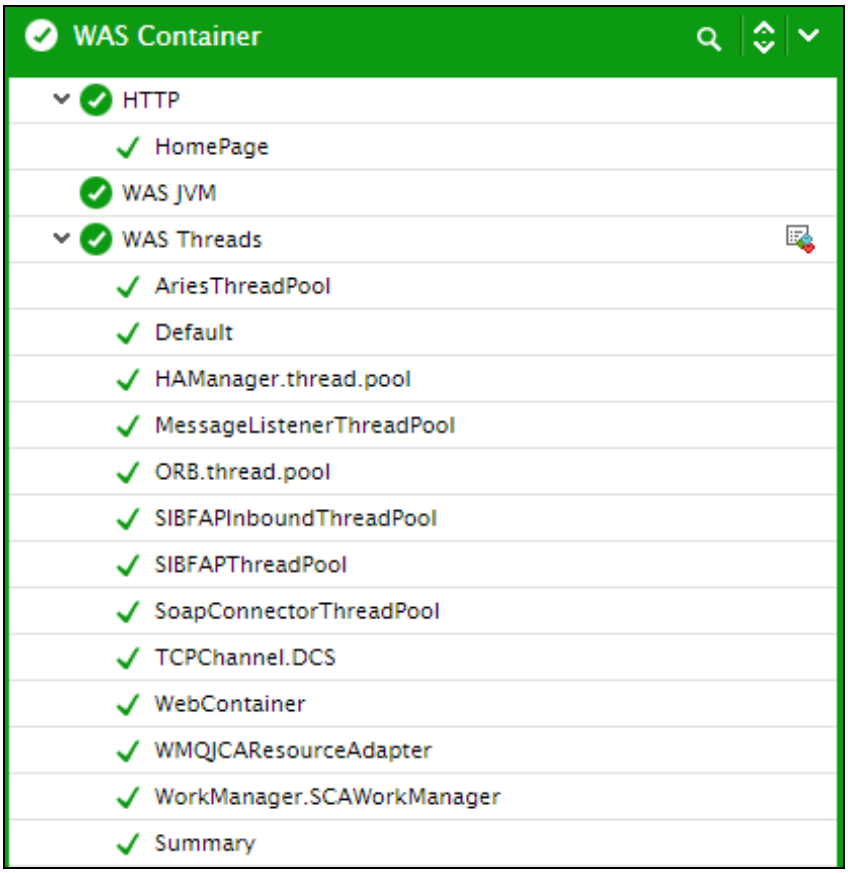

Figure 6.3: The tests associated with the WAS Container layer

6.2.1 WAS Cache Test

This test monitors the cache usage on the WebSphere server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each cache on the server instance being monitored

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
ServerHostName	Specify the host name of the application server instance being monitored.
AppPort	Specify the port number to be used for accessing the **egurkha** application that has been deployed on the server.
NodeName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the **Node** that corresponds to the application server instance being monitored, and specify that name against the NodeName parameter.
ServerName	Provide the name of the server instance being monitored in the **SERVERNAME** text box. To know the server name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the Name of the monitored server instance, and specify that name against the ServerName parameter. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the host name that corresponds to the connector port of the *Deployment Manager* of the cluster as the ServerName. To determine the ServerName in this case, do the following:

Parameters	Description
	• Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. Click on the **Details** button alongside the ports list. • If the SOAP port has been set as the connector port in your environment, then scroll down the page that appears next until you view the **Port Name**, **SOAP_CONNECTOR_ADDRESS**. Make note of the **Host** name that corresponds to this port name. If the RMI port is the connector port in your environment, then make note of the **Host** name that corresponds to the **Port** name, **BOOTSTRAP_ADDRESS**. • Specify this **Host** name as the ServerName.
ConnectorPort	The applications that are deployed on a server instance use the ConnectorPort for all internal communications with the application server. The connector port can be a SOAP port or an RMI port. The default connector port however, is the SOAP port. To know the connector port number, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on **Application Servers** within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. In the right pane, click on the server name link that corresponds to the server instance that is being monitored. • Doing so invokes the **Configuration** of the application server instance clicked

Parameters	Description
	on. Scroll down the **Configuration** tab page to view the **Communications** section. • Expand the **Ports** link in this section to view a list of ports. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the SOAP/RMI port of the *Deployment Manager* of the cluster as the ConnectorPort. To determine the ConnectorPort in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console. • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ ADDRESS** as the ConnectorPort.
SSL	Select **Yes** if **SSL** (Secured Socket Layer) is to be used to connect to the WebSphere server, and **No** if it is not.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid **USER** name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the **USER** text box should contain the default value 'none'.

Parameters	Description
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Client requests	Indicates the number of requests for cacheable objects that were generated by applications running on the application server, since the last measurement period.	Number	
Explicit invalidation	Indicates the number of explicit invalidations since the last measurement period.	Number	
Hits on memory	Indicates the number of requests for cacheable objects that were served from memory since the last measurement period.	Number	While direct disk reads cause an increase in processing overheads, memory reads are less expensive. Therefore, a high value of this measure is indicative of the good health of the server.
Hits on disk	Indicates the number of requests for cacheable objects that are served from the disk since the last measurement period.	Number	Reading from the disk is more expensive than reading from the cache. Therefore, ideally, this value should be low.
Memory cache entry	Indicates the number of in-memory cache entries since the last measurement period.	Number	

Measurement	Description	Measurement Unit	Interpretation
Max memory cache entry	Indicates the maximum number of in-memory cache entries.	Number	
Miss count:	Indicates the number of requests for cacheable objects that were not served from the cache since the last measurement period.	Number	Ideally, this value should be low.
Remote creations	Indicates the number of cache entries that were received from co-operating dynamic caches, since the last measurement period.	Number	
Remote hits	Indicates the number of requests for cacheable objects that were served from other JVMs within the replication domain, since the last measurement period.	Number	
Timeout invalidations	Indicates the number of cache entries that were removed from the memory and disk because of a timeout, since the last measurement period.	Number	

6.2.2 WAS JVM Test

This test monitors the usage of the WebSphere JVM heap.

Note:

To enable the test to collect metrics related to the garbage collection and thread activity, you should enable the Java Virtual Machine Tool Interface (JVMTI) of the WebSphere Server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each JVM being monitored

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
ServerHostName	Specify the host name of the application server instance being monitored.
AppPort	Specify the port number to be used for accessing the **egurkha** application that has been deployed on the server.
NodeName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the **Node** that corresponds to the application server instance being monitored, and specify that name against the NodeName parameter.
ServerName	Provide the name of the server instance being monitored in the **SERVERNAME** text box. To know the server name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the Name of the monitored server instance, and specify that name against the ServerName parameter. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the host name that corresponds to the connector port of the *Deployment*

Parameters	Description
	Manager of the cluster as the ServerName. To determine the ServerName in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. Click on the **Details** button alongside the ports list. • If the SOAP port has been set as the connector port in your environment, then scroll down the page that appears next until you view the **Port Name**, **SOAP_CONNECTOR_ADDRESS**. Make note of the **Host** name that corresponds to this port name. If the RMI port is the connector port in your environment, then make note of the **Host** name that corresponds to the **Port** name, **BOOTSTRAP_ADDRESS**. • Specify this **Host** name as the ServerName.
ConnectorPort	The applications that are deployed on a server instance use the ConnectorPort for all internal communications with the application server. The connector port can be a SOAP port or an RMI port. The default connector port however, is the SOAP port. To know the connector port number, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on **Application Servers** within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. In the right pane, click on the server name link that corresponds to the server instance that is being monitored.

Parameters	Description
	• Doing so invokes the **Configuration** of the application server instance clicked on. Scroll down the **Configuration** tab page to view the **Communications** section. • Expand the **Ports** link in this section to view a list of ports. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the SOAP/RMI port of the *Deployment Manager* of the cluster as the ConnectorPort. To determine the ConnectorPort in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console. • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ ADDRESS** as the ConnectorPort.
SSL	Select **Yes** if **SSL** (Secured Socket Layer) is to be used to connect to the WebSphere server, and **No** if it is not.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid **USER** name to login to the WebSphere server. If the WebSphere server does

Parameters	Description
	not require any authentication, then the **USER** text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Free memory	Indicates the amount of memory currently available in the JVM.	MB	A very low value of this measure is indicative of excessive memory utilization in the JVM.
Used memory	Indicates the amount of memory that has been currently utilized by the JVM.	MB	A high value of this measure indicates a heavy workload on the WebSphere application server. In such a case, you might want to consider increasing the JVM heap size.
GC count	Indicates the number of GC calls since the last measurement period.	Number	If adequate memory is not allotted to the JVM, then the value of this measure would be very high. A high value of this measure is indicative of a high frequency of GC. This is not a good sign, as GC, during its execution, has the tendency of suspending an application, and a high frequency of GC would only adversely impact the application's performance. To avoid this, it is recommended that you allot sufficient memory to the JVM.
Percent gc time	Indicates the percentage of time spent on GC.	Percent	If adequate memory is not allotted to the JVM, then the value of this measure would be very high. This is not a good sign, as GC, during its execution, has

Measurement	Description	Measurement Unit	Interpretation
			the tendency of suspending an application, and a high value of this measure would only adversely impact the application's performance. To avoid this, it is recommended that you allot sufficient memory to the JVM.
Percent heap used	Indicates the percentage of heap memory utilized by the JVM.	Percent	A high value of this measure indicates a heavy workload on the WebSphere application server. In such a case, you might want to consider increasing the JVM heap size.
Objects allocated	Indicates the number of objects allocated in the heap since the last measurement period.	Number	
Objects freed	Indicates the number of objects freed in the heap since the last measurement period.	Number	
Objects moved	Indicates the number of objects in the heap since the last measurement period.	Number	
Threads started	Indicates the number of threads started since the last measurement period.	Number	A high value is indicative of high server workload.
Threads ended	Indicates the number of threads that ended since the last measurement period.	Number	
Waits for lock	Indicates the number of times a thread waits for a lock, since the last measurement period.	Number	
Wait for lock time	Indicates the average time a thread waits for a lock.	Secs	

6.2.3 WAS Object Pools Test

This test reports critical statistics pertaining to the object pools on the WebSphere server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each Object Pool being monitored

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
ServerHostName	Specify the host name of the application server instance being monitored.
AppPort	Specify the port number to be used for accessing the **egurkha** application that has been deployed on the server.
NodeName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the **Node** that corresponds to the application server instance being monitored, and specify that name against the NodeName parameter.
ServerName	Provide the name of the server instance being monitored in the **SERVERNAME** text box. To know the server name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within.

Parameters	Description
	• A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the Name of the monitored server instance, and specify that name against the ServerName parameter. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the host name that corresponds to the connector port of the *Deployment Manager* of the cluster as the ServerName. To determine the ServerName in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. Click on the **Details** button alongside the ports list. • If the SOAP port has been set as the connector port in your environment, then scroll down the page that appears next until you view the **Port Name**, **SOAP_CONNECTOR_ADDRESS**. Make note of the **Host** name that corresponds to this port name. If the RMI port is the connector port in your environment, then make note of the **Host** name that corresponds to the **Port** name, **BOOTSTRAP_ADDRESS**. • Specify this **Host** name as the ServerName.
ConnectorPort	The applications that are deployed on a server instance use the ConnectorPort for all internal communications with the application server. The connector port can be a SOAP port or an RMI port. The default connector port however, is the SOAP port. To know the connector port number, do the following:

Parameters	Description
	• Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on **Application Servers** within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. In the right pane, click on the server name link that corresponds to the server instance that is being monitored. • Doing so invokes the **Configuration** of the application server instance clicked on. Scroll down the **Configuration** tab page to view the **Communications** section. • Expand the **Ports** link in this section to view a list of ports. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the SOAP/RMI port of the *Deployment Manager* of the cluster as the ConnectorPort. To determine the ConnectorPort in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console. • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be

Parameters	Description
	specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ ADDRESS** as the ConnectorPort.
SSL	Select **Yes** if **SSL** (Secured Socket Layer) is to be used to connect to the WebSphere server, and **No** if it is not.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid **USER** name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the **USER** text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Objects created	Indicates the number of new objects created by the object pool, since the last measurement period.	Number	
Objects allocated	Indicates the current count of objects allocated by the object pool.	Number	
Objects returned	Indicates the current count of objects returned to the pool.	Number	
Idle objects	Indicates the current count of idle objects.	Number	

6.2.4 WAS Threads Test

To optimize performance and at the same time to support concurrent accesses from users, the application server uses thread pools. It is critical to monitor a WebSphere server's thread pools on an ongoing basis. This test monitors the thread pools on a WebSphere server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each thread on the WebSphere application server

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
ServerHostName	Specify the host name of the application server instance being monitored.
AppPort	Specify the port number to be used for accessing the **egurkha** application that has been deployed on the server.
NodeName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the **Node** that corresponds to the application server instance being monitored, and specify that name against the NodeName parameter.
ServerName	Provide the name of the server instance being monitored in the **SERVERNAME** text box. To know the server name, do the following: • Login to the WebSphere Administrative Console.

Parameters	Description
	• Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the Name of the monitored server instance, and specify that name against the ServerName parameter. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the host name that corresponds to the connector port of the *Deployment Manager* of the cluster as the ServerName. To determine the ServerName in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. Click on the **Details** button alongside the ports list. • If the SOAP port has been set as the connector port in your environment, then scroll down the page that appears next until you view the **Port Name**, **SOAP_CONNECTOR_ADDRESS**. Make note of the **Host** name that corresponds to this port name. If the RMI port is the connector port in your environment, then make note of the **Host** name that corresponds to the **Port** name, **BOOTSTRAP_ADDRESS**. • Specify this **Host** name as the ServerName.
ConnectorPort	The applications that are deployed on a server instance use the ConnectorPort for all internal communications with the application server. The connector port can be a

Parameters	Description
	SOAP port or an RMI port. The default connector port however, is the SOAP port. To know the connector port number, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on **Application Servers** within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. In the right pane, click on the server name link that corresponds to the server instance that is being monitored. • Doing so invokes the **Configuration** of the application server instance clicked on. Scroll down the **Configuration** tab page to view the **Communications** section. • Expand the **Ports** link in this section to view a list of ports. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the SOAP/RMI port of the *Deployment Manager* of the cluster as the ConnectorPort. To determine the ConnectorPort in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console. • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node.

Parameters	Description
	• A list of ports will then appear. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ ADDRESS** as the ConnectorPort.
SSL	Select **Yes** if **SSL** (Secured Socket Layer) is to be used to connect to the WebSphere server, and **No** if it is not.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid **USER** name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the **USER** text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Active count	Indicates the number of threads that are currently active.	Number	A high value for this measure is indicative of a high load on this application and combined with the creation and destroy rates might give insights of the application pattern. This measure is also useful for determining usage trends. For example, it can show the time of day and the day of the week in which you usually reach peak thread count. In

Measurement	Description	Measurement Unit	Interpretation
			addition, the creation of too many threads can result in out of memory errors or thrashing. By watching this metric, you can reduce excessive memory consumption before it's too late.
Create count	Indicates the number of threads that were created since the last measurement period.	Number	A sudden increase in the value of this measure directly relates to an increase in the activity happening in this application.
Destroy count	Indicates the number of threads that were destroyed since the last measurement period.	Number	A decrease in the value of this measure indicates that the threads are being active for a long period of time, which might indicate anomalies within the application.
Pool size	Indicates the number of threads in the pool, currently.	Number	If the pool size is high and the number of active threads is low, it signifies that the threads are not being destroyed immediately after use.
Threads hung count	Indicates the number of concurrently stopped threads.	Number	

6.2.5 Web Service Test

A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. A complete web service is, therefore, any service that:

- Is available over the Internet or private (intranet) networks
- Uses a standardized XML messaging system
- Is not tied to any one operating system or programming language

- Is self-describing via a common XML grammar
- Is discoverable via a simple find mechanism

The basic web services platform is XML + HTTP. All the standard web services work using the following components:

- SOAP (Simple Object Access Protocol)
- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Services Description Language)

A web service enables communication among various applications by using open standards such as HTML, XML, WSDL, and SOAP. A web service takes the help of the following:

- XML to tag the data
- SOAP to transfer a message
- WSDL to describe the availability of service.

The following are the major uses of the Web Services:

- **Reusable application- components** : Often applications need repeated access to application- components like currency conversion, weather reports, or even language translation. In such cases, the web services can be used to offer the application-components as services with ease.
- **Connect existing software**: Web services can help to solve the interoperability problem by giving different applications a way to link their data. With Web services you can exchange data between different applications and different platforms. Any application can have a Web Service component. Web Services can be created regardless of programming language.

In certain environments, administrators are required to keep an eye on the web services that offer repeated access to the application-components i.e., operations so that the work load on the users using those application components can be minimized. If for some reason the web service takes too long to respond or is unavailable to cater to the needs of the users, then the users will be deprived of access to the application- components involved in that particular web service. To avoid such inconvenience caused to the users, administrators are required to continuously monitor the web services. The **Web Service** test helps administrators to perform this task perfectly. By continuosly monitoring each operation i.e., application component of a web service that is offered, using the SOAP commands, this test helps administrators identify the availability, response time and response code of the web service and quickly figure out discrepancies if any web service is deemed

unavailable. This way, the web services can be kept available round the clock thus helping the users perform their tasks without any difficulty.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each WebService:Operation i.e., application-component performed on the target server that is being monitored.

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WSDL URL	This test emulates a user accessing a specific web service(s) on the target server to determine the availability and responsiveness of the server. to enable this emulation, you need to configure the test with the url of the web service that it should access. specify this url against the WSDL URL parameter. if required, you can even configure multiple WSDL URLs - one each for every web service that the test should attempt to access. if each WSDL URL configured requires special permissions for logging in, then, you need to configure the test with separate credentials for logging into every WSDL URL. likewise, you need to provide instructions to the test on how to validate the content returned by every WSDL URL, and also set an encoding format for each wsdl url. to enable administrators to easily configure the above per WSDL URL, eg enterprise provides a special interface. to access this interface, click on the encircled '+' button alongside the url text box in the test configuration page. alternatively, you can even click on the encircled '+' button adjacent to the WSDL URL parameter in the test configuration page. to know how to use this special interface, refer to section 6.2.5.1.
Operations	Once the WSDL URL(s) are specified, the operations that are offered by the web services and those that are to be monitored have to be configured. To select the required operations for monitoring, eG Enterprise provides a special interface. TO access this interface, click on the encircled '+' button alongside the Operations text box in the test configuration page. Alternatively, you can even click on the encircled '+' button adjacent to the OPERATIONS parameter in the test configuration page. To know how to use this special interface, refer to section 6.2.5.1.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the server. If there is no response from the server beyond the configured duration, the test will timeout. By default, this is set to 30 seconds.

Parameters	Description
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option. To disable the capability, click on the Off option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
WSDL url availability:	Indicates whether the web service was able to respond successfully to the query made by the test.	Percent	Availability failures could be caused by several factors such as the web service process(es) being down, the web service being misconfigured, a network failure, etc. Temporary unavailability may also occur if the web service is overloaded. Availability is determined based on the response code returned by the service. A response code between 200 to 300 indicates that the service is available.
WSDL response time:	Indicates the time taken by the eG agent to get the configured web service.	Secs	Response time being high denotes a problem. Poor response times may be due to the service being overloaded or misconfigured. If the URL accessed involves the generation of dynamic content by the service, backend problems (e.g., an overload at the application server or a database failure) can also result in an increase in response time.
Port status:	Indicates whether/not the		The values reported by this measure

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td></td><td>port of the web server is reachable.</td><td></td><td>and the corresponding numeric equivalents are listed in the table below:
<table><tr><th>Measure Values</th><th>Numeric Values</th></tr><tr><td>Yes</td><td>1</td></tr><tr><td>No</td><td>0</td></tr></table>
Note:
By default, this measure reports the above-mentioned Measure Values to indicate whether the server has been rebooted or not. In the graph of this measure however, the Measure Values are represented using the numeric equivalents only.</td></tr>
<tr><td>TCP connection availability:</td><td>Indicates whether the test managed to establish a TCP connection to the server.</td><td>Percent</td><td>Failure to establish a TCP connection may imply that either the web server process is not up, or that the process is not operating correctly. In some cases of extreme overload, the failure to establish a TCP connection may be a transient condition. As the load subsides, the server may start functioning properly again.</td></tr>
<tr><td>TCP connect time:</td><td>This measure quantifies the time for establishing a TCP connection to the web server host.</td><td>Secs</td><td>Typically, the TCP connection establishment must be very small (of the order of a few milliseconds). Since TCP connection establishment is handled at the OS-level, rather than by the application, an increase in this value signifies a system-level bottleneck on the host that supports the web server.</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
Server response time:	Indicates the time period between when the connection was established and when the web server sent back a response header to the client.	Secs	While the total response time may depend on several factors, this measure is typically, a very good indicator of a server bottleneck (e.g., because all the available server threads or processes are in use).
Response code:	The response code returned by the web server for the simulated request.	Number	A value between 200 and 300 indicates a good response. A 4xx value indicates a problem with the requested content (eg., page not found). A 5xx value indicates a server error.
Service availability:	Indicates whether/not the web service is available.	Percent	A value of 100 indicates that the web service is available and a value of 0 indicates that the web service is not available.
Operation status:	Indicates whether/not the configured operation is present in the web service.		This measure will not report metrics if the OPERATION parameter in the test configuration page is none in the test configuration page.
Operation Content length:	Indicates the response code returned by the server for the simulated request.	Number	A value between 200 and 300 indicates a good response. A 4xx value indicates a problem with the requested content (e.g., page not found). A 5xx value indicates a server error. This measure will not report metrics if the OPERATION parameter in the test configuration page is none or if an invalid Value is specified or if the Value is not specified in the HTML View tab while configuring the operation for monitoring in the test configuration page.
Operation Content validity:	This measure validates whether the operation was successful in executing the request made to it.	Percent	A value of 100% indicates that the content returned by the test is valid. A value of 0% indicates that the content

Measurement	Description	Measurement Unit	Interpretation
			may not be valid. This capability for content validation is especially important for multi-tier web applications. For example, a user may not be able to login to the web site but the server may reply back with a valid HTML page where in the error message, say, “Invalid Login” is reported. In this case, the availability will be 100 % (since we got a valid HTML response). If the test is configured such that the content parameter should exclude the string “Invalid Login”, in the above scenario content validity would have a value 0. This measure will not report metrics if the OPERATION parameter in the test configuration page is none or if an invalid Value is specified or if the Value is not specified in the HTML View tab while configuring the operation for monitoring in the test configuration page.
Operation execution time:	Indicates the time taken to invoke the configured operation in the web service.	Secs	This measure will not report metrics if the OPERATION parameter in the test configuration page is none or if an invalid Value is specified or if the Value is not specified in the HTML View tab while configuring the operation for monitoring in the test configuration page.

6.2.5.1 Configuring Multiple WSDL URLs for Monitoring

In order to enable the eG agent to connect to multiple WSDL URLs and pull out the required metrics from them, the eG administrative interface provides a special page using which different WSDL URLs and their corresponding operations that need to be monitored can be specified. To configure the WSDL URLs, do the following:

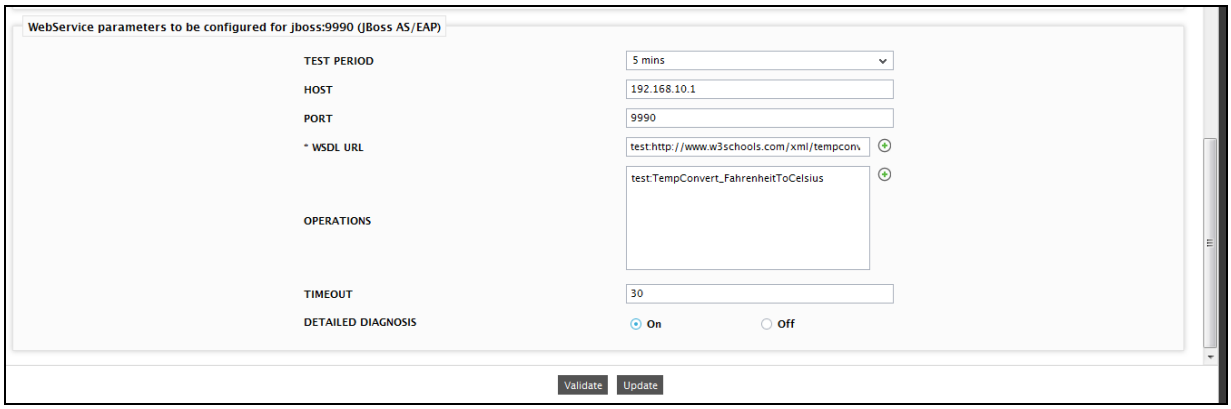

Figure 6.4: Configuring the WebService test

1. Click on the encircled '+' button alongside the **WSDL URL** text box. Figure 6.5 will then appear.

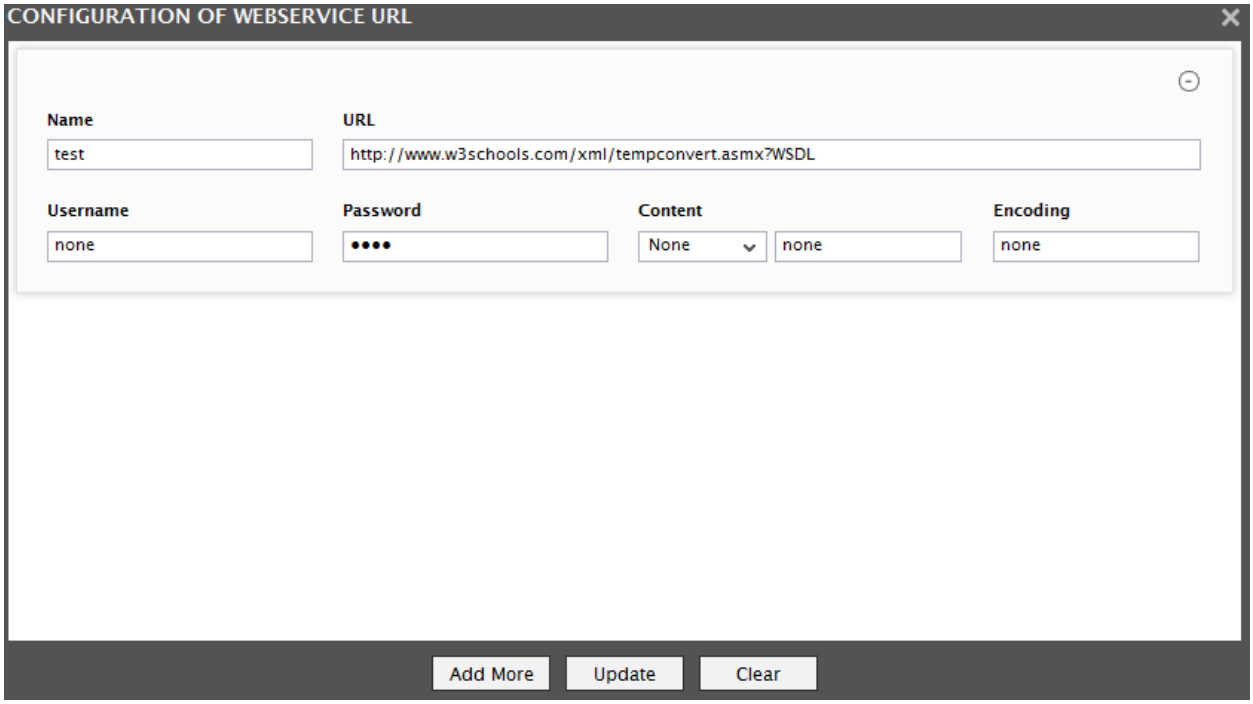

Figure 6.5: Figure 3.10: The WebService URL Configuration page

2. Specify the following in Figure 6.5:

   - **Name:** Specify a unique name by which the WSDL URL you will be specifiying shortly will be referred to across the eG user interface. This is the name that will appear as the descriptor of this test.

- **URL:** Enter the WSDL URL of the web service that this test should access.
- **Username** and **Password:** These parameters are to be set only if a specific user name / password has to be specified to login to the web service (i.e., WSDL URL) that you have configured for monitoring. In this case, provide valid login credentials using the **Username** and **Password** text boxes. If the server on which **WebService** test executes supports 'Anonymous user access', then these parameters will take either of the following values:
  - A valid **Username** and **Password** for the configured **WSDL URL**
  - none in both the **Username** and **Password** text boxes of the configured WSDL URL, if no user authorization is required
  - Some servers however, support NTLM (Integrated Windows) authentication, where valid login credentials are mandatory. In other words, a none specification will not be supported by such servers. Therefore, in this case, against each configured **WSDL URL**, you will have to provide a valid **Username** in the format: domainname\username, followed by a valid **Password**.
  - Please be sure to check if your web service requires HTTP authentication while configuring this parameter. HTTP authentication typically involves a separate pop-up window when you try to access the page. Many services use HTTP POST for obtaining the user name and password and validating the user login. In such cases, the Username and Password have to be provided as part of the POST information and NOT as part of the **CREDENTIALS** specification for the WebService test.
- **Content**: The **Content** parameter has to be configured with an instruction:value pair that will be used to validate the content being returned by the test. If the **Content** value is None, no validation is performed. On the other hand, if you pick the Include option from the **Content** list, it indicates to the test that for the content returned by the web server to be valid, the content must include the specified value (a simple string search is done in this case). This value should be specified in the adjacent text box. Similarly, if the Exclude option is chosen from the **Content** drop-down, it indicates to the test that the server's output is valid if it does not contain the value specified in the adjacent text box. The Include or Exclude value you specify in the text box can include wildcard characters. For example, an Include instruction can be *Home page*.
- **Encoding**: Sometimes the eG agent has to parse the **WSDL URL** content with specific encoding other than the default (ISO-8859-1) encoding. In such a case, specify the type of encoding using which the eG agent can parse the **WSDL URL** content in the **Encoding** text box. By default, this value is none.

3. Similarly, you can add multiple URL specifications by clicking the **Add More** button. To remove a WSDL URL specification, click on the encircled '-' button corresponding to it. To clear all **WSDL URL** specifications, click the **Clear** button. To update all the changes you made, click the **Update** button.
4. Once **Update** is clicked, you will return to the test configuration page as shown in Figure 6.4. The **WSDL URL** text box in the test configuration page will display just the **Name**s - i.e., the unique display names - that you may have configured for the multiple WSDL URLs, as a comma-separated list. To view the complete WSDL URL specification, click the encircled '+' button alongside the **WSDL URL** text box, once again.

6.2.5.2 Configuring Multiple Operations for Monitoring - WebServiceTest

By default, the **WebServiceTest** test will be configured with the WSDL URLs that offer the web services that are to be monitored. To configure the operations that are offered by the WSDL URLs, do the following:

1. Click on the encircled '+' button alongside the **OPERATIONS** text box as shown in Figure 6.4. Figure 6.6 will then appear.

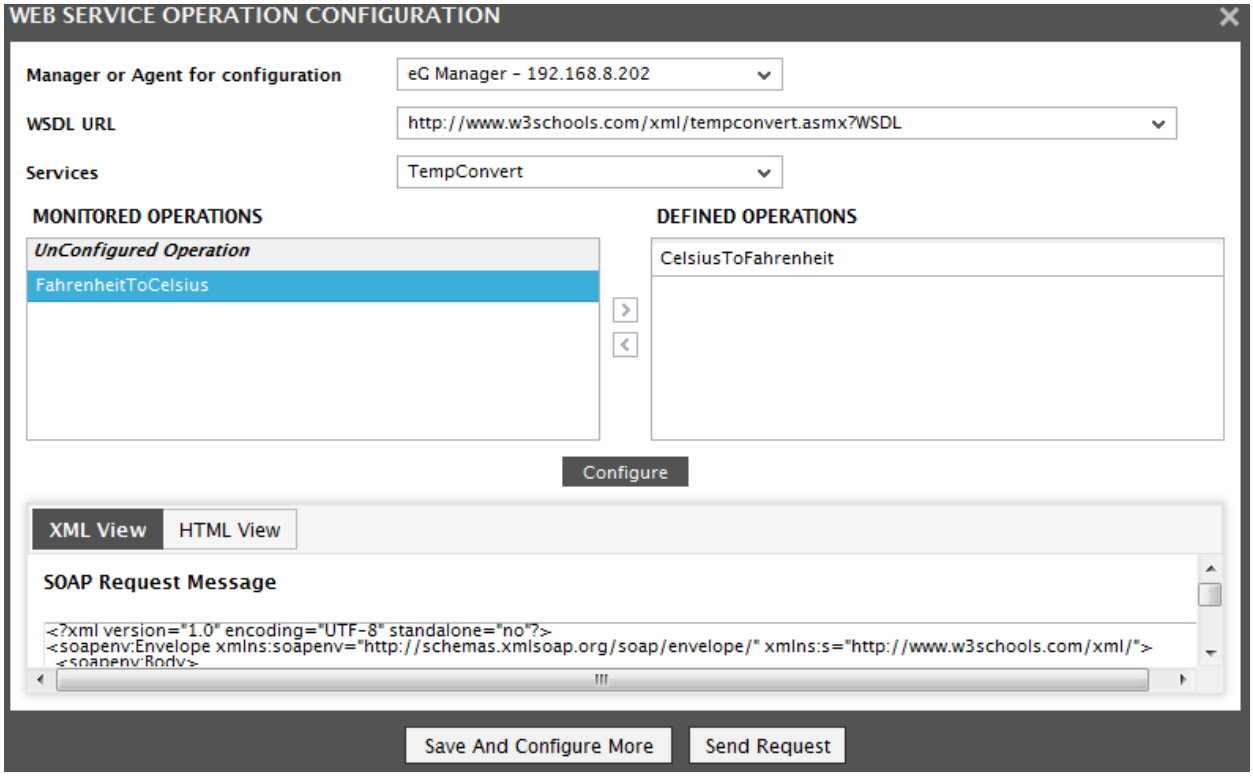

Figure 6.6: Figure 3.11: Configuring the Web Service Operation

2. Specify the following in Figure 6.6:

- Manager/Agent for accessing WSDL URL: Select the eG agent or the eG Manager that is authorized to access the configured WSDL URL from this list.
- WSDL URL: Once the eG agent/eG Manager is chosen from the Manager/Agent for accessing WSDL URL list, this list will be populated automatically with all the WSDL URLs specified in the **WSDL URL** text box (See Figure 6.4). Select the **WSDL URL** of your choice from this list.
- Services: The web services offered by the chosen WSDL URL will then be populated in this list. Select a service of your choice from this list.
- The operations that are offered by the chosen service will then be populated in the **DEFINED OPERATIONS** list. To monitor a chosen operation, select the operation and click the < button. This will move the chosen operation to the **MONITORED OPERATIONS** list.
- Click the Configure button to save the changes.
- The eG agent uses SOAP requests to obtain the necessary metrics from the web service. Once the operation is configured, the XML View of the SOAP Request corresponding to the chosen operation will be generated and listed in the **XML View** tab. Likewise, the **HTML View** tab lists the **SOAP Parameter** that is passed to collect the required metrics for the chosen operation.
- To obtain operation-level statistics, it is important to specify a valid value in the VALUE text box of the HTML View tab as shown in Figure 6.6. Each time the test is executed, this value will be provided as an input to the chosen operation.

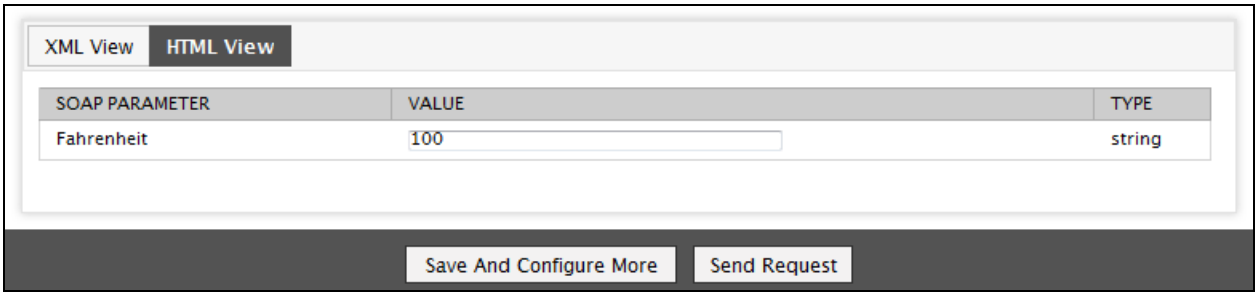

Figure 6.7: Figure 3.12: Specifying the value for the chosen operation

- Click the **Save and Cofigure More** button to save the changes made.
- If you wish to verify if the **VALUE** specified in the HTML View tab is valid, then you can do so by clicking the **Send Request** button. Figure 6.7 will then appear. If the value specified in the **VALUE**

text box is indeed valid, then the operation will be performed on the value and the result will be specified. For example, if your chosen operation is FahrenheittoCelsius, the SOAP Parameter is Farenheit and the value that you wish to convert is 100, the result will be specified in the **WEB SERVICE RESPONSE** pop up window as below: <FahrenheitToCelsiusResult>37.7777777777778</FahrenheitToCelsiusResult>

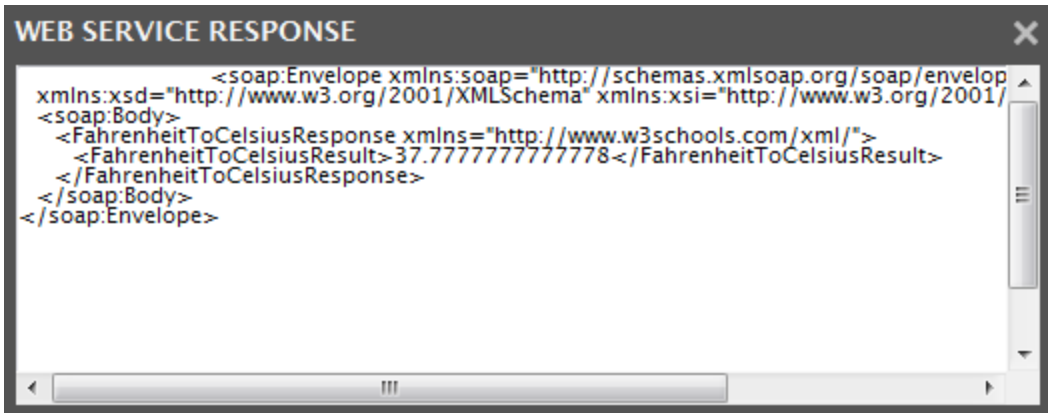

Figure 6.8: Figure 3.13: The value that appears when the operation is performed successfully

- If you have specified an invalid value, then a message as follows will be displayed in the pop up window:
  <FahrenheitToCelsiusResult>Error</FahrenheitToCelsiusResult>

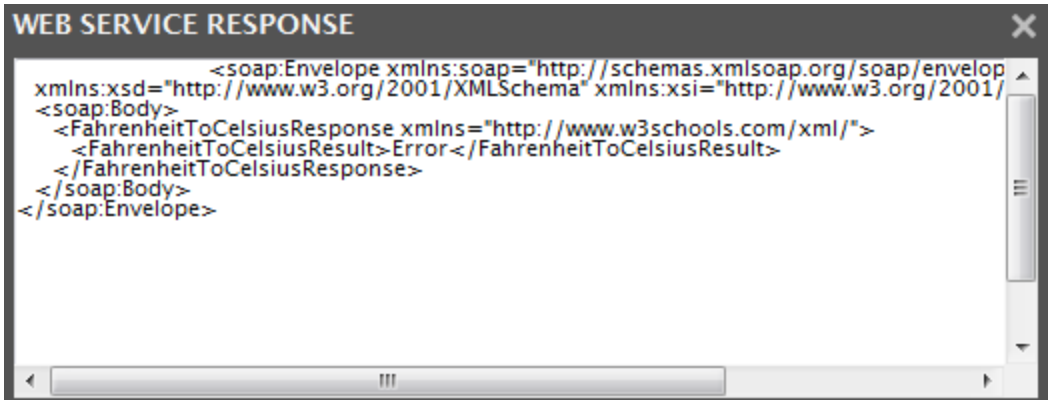

Figure 6.9: Figure 3.14: An Error appearing during value conversion

- If you do not specify a **VALUE** or specify an invalid value, operation-level statistics will not be collected by the eG agent and such metrics will not be available in the eG monitoring interface.

3. Similarly, you can configure multiple Operations by clicking the **Configure** button in Figure 6.6. To remove an operation, select the operation from the **MONITORED OPERATION** list and click the > button.
4. Once **Save and Configure More** button is clicked, you will return to the test configuration page (see Figure 6.6). The **OPERATIONS** text box in the test configuration page will display just the operations, as a comma-separated list. To view the complete operation specification, click the encircled '+' button alongside the **OPERATIONS** text box, once again.

6.2.6 Web Service Test

A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. A complete web service is, therefore, any service that:

- Is available over the Internet or private (intranet) networks
- Uses a standardized XML messaging system
- Is not tied to any one operating system or programming language
- Is self-describing via a common XML grammar
- Is discoverable via a simple find mechanism

The basic web services platform is XML + HTTP. All the standard web services work using the following components:

- SOAP (Simple Object Access Protocol)
- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Services Description Language)

A web service enables communication among various applications by using open standards such as HTML, XML, WSDL, and SOAP. A web service takes the help of the following:

- XML to tag the data
- SOAP to transfer a message
- WSDL to describe the availability of service.

The following are the major uses of the Web Services:

- **Reusable application- components** : Often applications need repeated access to application- components like currency conversion, weather reports, or even language translation. In such cases, the web services can be used to offer the application-components as services with ease.
- **Connect existing software**: Web services can help to solve the interoperability problem by giving different applications a way to link their data. With Web services you can exchange data between different applications and different platforms. Any application can have a Web Service component. Web Services can be created regardless of programming language.

In certain environments, administrators are required to keep an eye on the web services that offer repeated access to the application-components i.e., operations so that the work load on the users using those application components can be minimized. If for some reason the web service takes too long to respond or is unavailable to cater to the needs of the users, then the users will be deprived of access to the application- components involved in that particular web service. To avoid such inconvenience caused to the users, administrators are required to continuously monitor the web services. The **Web Service** test helps administrators to perform this task perfectly. By continuosly monitoring each operation i.e., application component of a web service that is offered, using the SOAP commands, this test helps administrators identify the availability, response time and response code of the web service and quickly figure out discrepancies if any web service is deemed unavailable. This way, the web services can be kept available round the clock thus helping the users perform their tasks without any difficulty.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each WebService:Operation i.e., application-component performed on the target server that is being monitored.

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WSDL URL	This test emulates a user accessing a specific web service(s) on the target server to determine the availability and responsiveness of the server. to enable this emulation, you need to configure the test with the url of the web service that it should access.

Parameters	Description
	specify this url against the WSDL URL parameter. if required, you can even configure multiple WSDL URLs - one each for every web service that the test should attempt to access. if each WSDL URL configured requires special permissions for logging in, then, you need to configure the test with separate credentials for logging into every WSDL URL. likewise, you need to provide instructions to the test on how to validate the content returned by every WSDL URL, and also set an encoding format for each wsdl url. to enable administrators to easily configure the above per WSDL URL, eg enterprise provides a special interface. to access this interface, click on the encircled '+' button alongside the url text box in the test configuration page. alternatively, you can even click on the encircled '+' button adjacent to the WSDL URL parameter in the test configuration page. to know how to use this special interface, refer to section 6.2.6.1.
Operations	Once the WSDL URL(s) are specified, the operations that are offered by the web services and those that are to be monitored have to be configured. To select the required operations for monitoring, eG Enterprise provides a special interface. TO access this interface, click on the encircled '+' button alongside the Operations text box in the test configuration page. Alternatively, you can even click on the encircled '+' button adjacent to the OPERATIONS parameter in the test configuration page. To know how to use this special interface, refer to section 6.2.6.1.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the server. If there is no response from the server beyond the configured duration, the test will timeout. By default, this is set to 30 seconds.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option. To disable the capability, click on the Off option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>WSDL url availability:</td><td>Indicates whether the web service was able to respond successfully to the query made by the test.</td><td>Percent</td><td>Availability failures could be caused by several factors such as the web service process(es) being down, the web service being misconfigured, a network failure, etc. Temporary unavailability may also occur if the web service is overloaded. Availability is determined based on the response code returned by the service. A response code between 200 to 300 indicates that the service is available.</td></tr>
<tr><td>WSDL response time:</td><td>Indicates the time taken by the eG agent to get the configured web service.</td><td>Secs</td><td>Response time being high denotes a problem. Poor response times may be due to the service being overloaded or misconfigured. If the URL accessed involves the generation of dynamic content by the service, backend problems (e.g., an overload at the application server or a database failure) can also result in an increase in response time.</td></tr>
<tr><td>Port status:</td><td>Indicates whether/not the port of the web server is reachable.</td><td></td><td>The values reported by this measure and the corresponding numeric equivalents are listed in the table below:
<table><tr><th>Measure Values</th><th>Numeric Values</th></tr><tr><td>Yes</td><td>1</td></tr><tr><td>No</td><td>0</td></tr></table>
Note:
By default, this measure reports the above-mentioned Measure Values to</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			indicate whether the server has been rebooted or not. In the graph of this measure however, the **Measure Value**s are represented using the numeric equivalents only.
TCP connection availability:	Indicates whether the test managed to establish a TCP connection to the server.	Percent	Failure to establish a TCP connection may imply that either the web server process is not up, or that the process is not operating correctly. In some cases of extreme overload, the failure to establish a TCP connection may be a transient condition. As the load subsides, the server may start functioning properly again.
TCP connect time:	This measure quantifies the time for establishing a TCP connection to the web server host.	Secs	Typically, the TCP connection establishment must be very small (of the order of a few milliseconds). Since TCP connection establishment is handled at the OS-level, rather than by the application, an increase in this value signifies a system-level bottleneck on the host that supports the web server.
Server response time:	Indicates the time period between when the connection was established and when the web server sent back a response header to the client.	Secs	While the total response time may depend on several factors, this measure is typically, a very good indicator of a server bottleneck (e.g., because all the available server threads or processes are in use).
Response code:	The response code returned by the web server for the simulated request.	Number	A value between 200 and 300 indicates a good response. A 4xx value indicates a problem with the requested content (eg., page not found). A 5xx value indicates a server error.
Service availability:	Indicates whether/not the web service is available.	Percent	A value of 100 indicates that the web service is available and a value of 0

Measurement	Description	Measurement Unit	Interpretation
			indicates that the web service is not available.
Operation status:	Indicates whether/not the configured operation is present in the web service.		This measure will not report metrics if the OPERATION parameter in the test configuration page is none in the test configuration page.
Operation Content length:	Indicates the response code returned by the server for the simulated request.	Number	A value between 200 and 300 indicates a good response. A 4xx value indicates a problem with the requested content (e.g., page not found). A 5xx value indicates a server error. This measure will not report metrics if the OPERATION parameter in the test configuration page is none or if an invalid Value is specified or if the Value is not specified in the HTML View tab while configuring the operation for monitoring in the test configuration page.
Operation Content validity:	This measure validates whether the operation was successful in executing the request made to it.	Percent	A value of 100% indicates that the content returned by the test is valid. A value of 0% indicates that the content may not be valid. This capability for content validation is especially important for multi-tier web applications. For example, a user may not be able to login to the web site but the server may reply back with a valid HTML page where in the error message, say, “Invalid Login” is reported. In this case, the availability will be 100 % (since we got a valid HTML response). If the test is configured such that the content parameter should exclude the string “Invalid Login”, in the above scenario content validity would have a value 0.

Measurement	Description	Measurement Unit	Interpretation
			This measure will not report metrics if the OPERATION parameter in the test configuration page is none or if an invalid Value is specified or if the Value is not specified in the HTML View tab while configuring the operation for monitoring in the test configuration page.
Operation execution time:	Indicates the time taken to invoke the configured operation in the web service.	Secs	This measure will not report metrics if the OPERATION parameter in the test configuration page is none or if an invalid Value is specified or if the Value is not specified in the HTML View tab while configuring the operation for monitoring in the test configuration page.

6.2.6.1 Configuring Multiple WSDL URLs for Monitoring

In order to enable the eG agent to connect to multiple WSDL URLs and pull out the required metrics from them, the eG administrative interface provides a special page using which different WSDL URLs and their corresponding operations that need to be monitored can be specified. To configure the WSDL URLs, do the following:

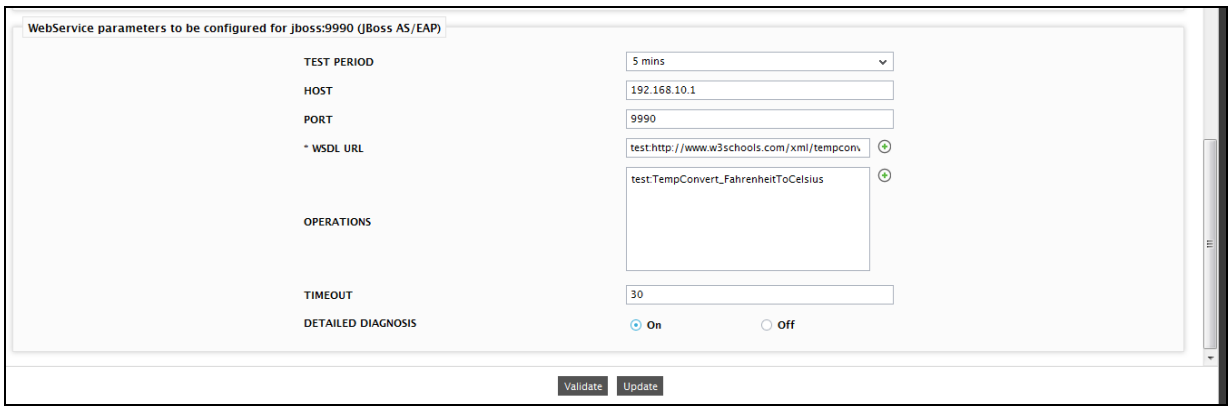

Figure 6.10: Configuring the WebService test

1. Click on the encircled '+' button alongside the **WSDL URL** text box. Figure 6.11 will then appear.

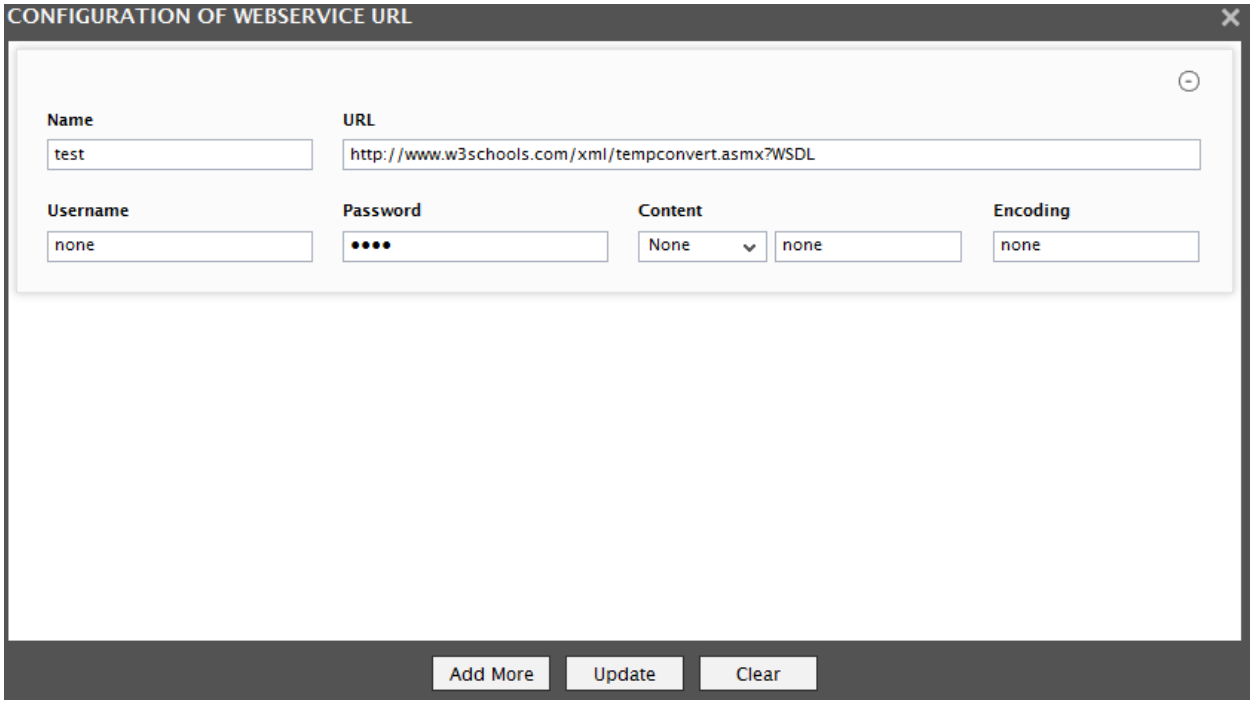

Figure 6.11: Figure 3.10: The WebService URL Configuration page

2. Specify the following in Figure 6.11:

- **Name:** Specify a unique name by which the WSDL URL you will be specifiying shortly will be referred to across the eG user interface. This is the name that will appear as the descriptor of this test.
- **URL:** Enter the WSDL URL of the web service that this test should access.
- **Username** and **Password:** These parameters are to be set only if a specific user name / password has to be specified to login to the web service (i.e., WSDL URL) that you have configured for monitoring. In this case, provide valid login credentials using the **Username** and **Password** text boxes. If the server on which **WebService** test executes supports 'Anonymous user access', then these parameters will take either of the following values:
  - A valid **Username** and **Password** for the configured **WSDL URL**
  - none in both the **Username** and **Password** text boxes of the configured WSDL URL, if no user authorization is required
  - Some servers however, support NTLM (Integrated Windows) authentication, where valid login credentials are mandatory. In other words, a none specification will not be supported

by such servers. Therefore, in this case, against each configured **WSDL URL**, you will have to provide a valid **Username** in the format: domainname\username, followed by a valid **Password**.

- Please be sure to check if your web service requires HTTP authentication while configuring this parameter. HTTP authentication typically involves a separate pop-up window when you try to access the page. Many services use HTTP POST for obtaining the user name and password and validating the user login. In such cases, the Username and Password have to be provided as part of the POST information and NOT as part of the **CREDENTIALS** specification for the WebService test.

- **Content**: The **Content** parameter has to be configured with an instruction:value pair that will be used to validate the content being returned by the test. If the **Content** value is None, no validation is performed. On the other hand, if you pick the Include option from the **Content** list, it indicates to the test that for the content returned by the web server to be valid, the content must include the specified value (a simple string search is done in this case). This value should be specified in the adjacent text box. Similarly, if the Exclude option is chosen from the **Content** drop-down, it indicates to the test that the server's output is valid if it does not contain the value specified in the adjacent text box. The Include or Exclude value you specify in the text box can include wildcard characters. For example, an Include instruction can be *Home page*.

- **Encoding**: Sometimes the eG agent has to parse the **WSDL URL** content with specific encoding other than the default (ISO-8859-1) encoding. In such a case, specify the type of encoding using which the eG agent can parse the **WSDL URL** content in the **Encoding** text box. By default, this value is none.

3. Similarly, you can add multiple URL specifications by clicking the **Add More** button. To remove a WSDL URL specification, click on the encircled '-' button corresponding to it. To clear all **WSDL URL** specifications, click the **Clear** button. To update all the changes you made, click the **Update** button.

4. Once **Update** is clicked, you will return to the test configuration page as shown in Figure 6.10. The **WSDL URL** text box in the test configuration page will display just the **Name**s - i.e., the unique display names - that you may have configured for the multiple WSDL URLs, as a comma-separated list. To view the complete WSDL URL specification, click the encircled '+' button alongside the **WSDL URL** text box, once again.

6.2.6.2 Configuring Multiple Operations for Monitoring - WebServiceTest

By default, the **WebServiceTest** test will be configured with the WSDL URLs that offer the web services that are to be monitored. To configure the operations that are offered by the WSDL URLs,

do the following:

1. Click on the encircled '+' button alongside the **OPERATIONS** text box as shown in Figure 6.10. Figure 6.12 will then appear.

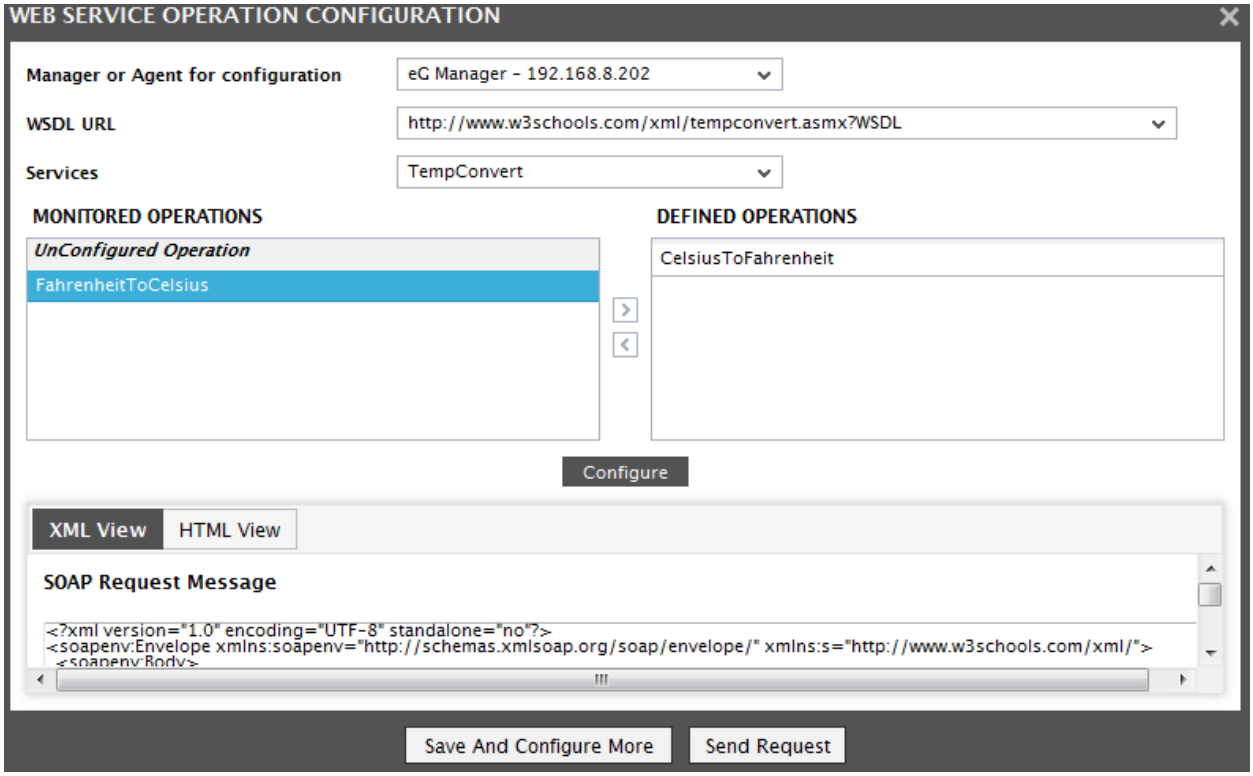

Figure 6.12: Figure 3.11: Configuring the Web Service Operation

2. Specify the following in Figure 6.12:

   - Manager/Agent for accessing WSDL URL: Select the eG agent or the eG Manager that is authorized to access the configured WSDL URL from this list.
   - WSDL URL: Once the eG agent/eG Manager is chosen from the Manager/Agent for accessing WSDL URL list, this list will be populated automatically with all the WSDL URLs specified in the **WSDL URL** text box (See Figure 6.10). Select the **WSDL URL** of your choice from this list.
   - Services: The web services offered by the chosen WSDL URL will then be populated in this list. Select a service of your choice from this list.

- The operations that are offered by the chosen service will then be populated in the **DEFINED OPERATIONS** list. To monitor a chosen operation, select the operation and click the < button. This will move the chosen operation to the **MONITORED OPERATIONS** list.
- Click the Configure button to save the changes.
- The eG agent uses SOAP requests to obtain the necessary metrics from the web service. Once the operation is configured, the XML View of the SOAP Request corresponding to the chosen operation will be generated and listed in the **XML View** tab. Likewise, the **HTML View** tab lists the **SOAP Parameter** that is passed to collect the required metrics for the chosen operation.
- To obtain operation-level statistics, it is important to specify a valid value in the VALUE text box of the HTML View tab as shown in Figure 6.12. Each time the test is executed, this value will be provided as an input to the chosen operation.

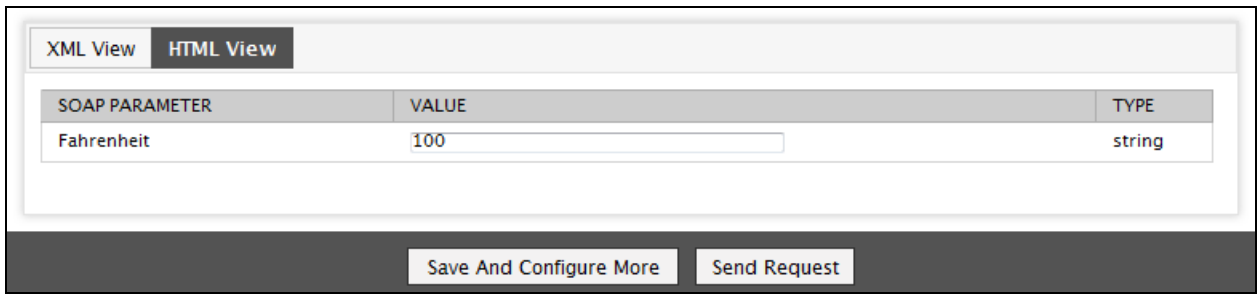

Figure 6.13: Figure 3.12: Specifying the value for the chosen operation

- Click the **Save and Cofigure More** button to save the changes made.
- If you wish to verify if the **VALUE** specified in the HTML View tab is valid, then you can do so by clicking the **Send Request** button. Figure 6.13 will then appear. If the value specified in the **VALUE** text box is indeed valid, then the operation will be performed on the value and the result will be specified. For example, if your chosen operation is FahrenheittoCelsius, the SOAP Parameter is Farenheit and the value that you wish to convert is 100, the result will be specified in the **WEB SERVICE RESPONSE** pop up window as below: <FahrenheitToCelsiusResult>37.7777777777778</FahrenheitToCelsiusResult>

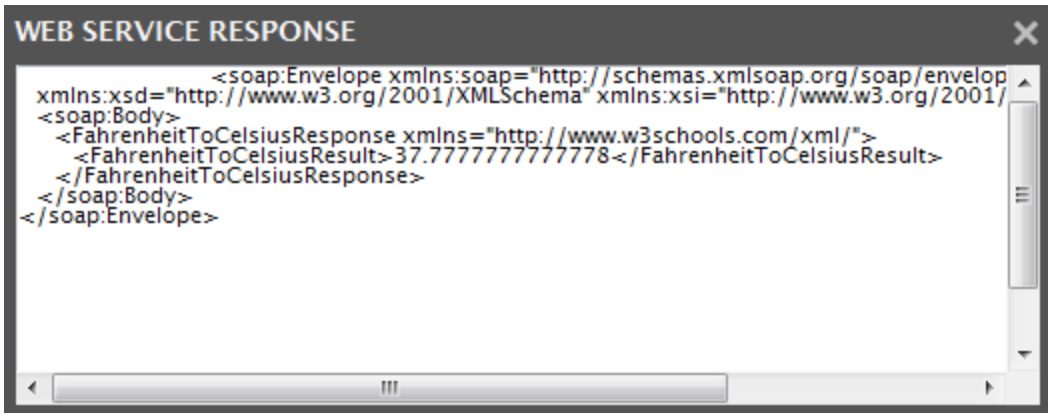

Figure 6.14: Figure 3.13: The value that appears when the operation is performed successfully

- If you have specified an invalid value, then a message as follows will be displayed in the pop up window:
  <FahrenheitToCelsiusResult>Error</FahrenheitToCelsiusResult>

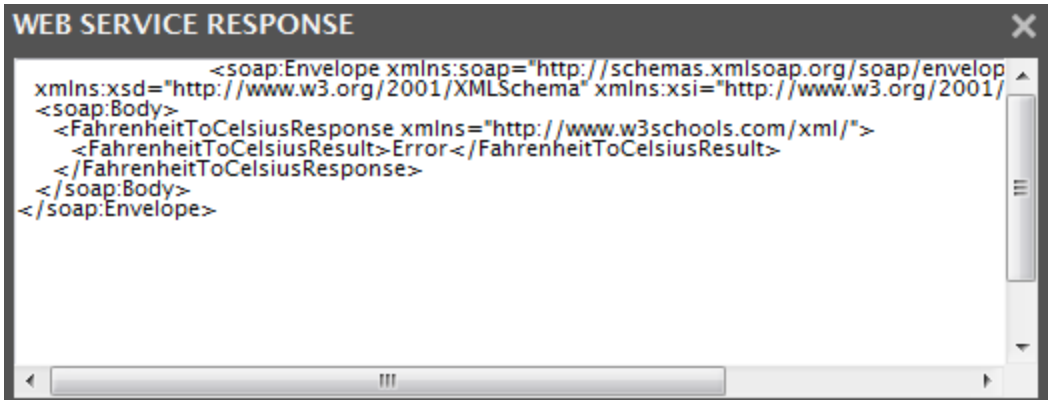

Figure 6.15: Figure 3.14: An Error appearing during value conversion

- If you do not specify a **VALUE** or specify an invalid value, operation-level statistics will not be collected by the eG agent and such metrics will not be available in the eG monitoring interface.

3. Similarly, you can configure multiple Operations by clicking the **Configure** button in Figure 6.12. To remove an operation, select the operation from the **MONITORED OPERATION** list and click the > button.
4. Once **Save and Configure More** button is clicked, you will return to the test configuration page (see Figure 6.12). The **OPERATIONS** text box in the test configuration page will display just the operations, as a comma-separated list. To view the complete operation specification, click the encircled '+' button alongside the **OPERATIONS** text box, once again.

6.2.7 Web Service Test

A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. A complete web service is, therefore, any service that:

- Is available over the Internet or private (intranet) networks
- Uses a standardized XML messaging system
- Is not tied to any one operating system or programming language
- Is self-describing via a common XML grammar
- Is discoverable via a simple find mechanism

The basic web services platform is XML + HTTP. All the standard web services work using the following components:

- SOAP (Simple Object Access Protocol)
- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Services Description Language)

A web service enables communication among various applications by using open standards such as HTML, XML, WSDL, and SOAP. A web service takes the help of the following:

- XML to tag the data
- SOAP to transfer a message
- WSDL to describe the availability of service.

The following are the major uses of the Web Services:

- **Reusable application- components** : Often applications need repeated access to application- components like currency conversion, weather reports, or even language translation. In such cases, the web services can be used to offer the application-components as services with ease.
- **Connect existing software**: Web services can help to solve the interoperability problem by giving different applications a way to link their data. With Web services you can exchange data between different applications and different platforms. Any application can have a Web Service component. Web Services can be created regardless of programming language.

In certain environments, administrators are required to keep an eye on the web services that offer repeated access to the application-components i.e., operations so that the work load on the users using those application components can be minimized. If for some reason the web service takes too long to respond or is unavailable to cater to the needs of the users, then the users will be deprived of access to the application-components involved in that particular web service. To avoid such inconvenience caused to the users, administrators are required to continuously monitor the web services. The **Web Service** test helps administrators to perform this task perfectly. By continuosly monitoring each operation i.e., application component of a web service that is offered, using the SOAP commands, this test helps administrators identify the availability, response time and response code of the web service and quickly figure out discrepancies if any web service is deemed unavailable. This way, the web services can be kept available round the clock thus helping the users perform their tasks without any difficulty.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each WebService:Operation i.e., application-component performed on the target server that is being monitored.

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
WSDL URL	This test emulates a user accessing a specific web service(s) on the target server to determine the availability and responsiveness of the server. to enable this emulation, you need to configure the test with the url of the web service that it should access. specify this url against the WSDL URL parameter. if required, you can even configure multiple WSDL URLs - one each for every web service that the test should attempt to access. if each WSDL URL configured requires special permissions for logging in, then, you need to configure the test with separate credentials for logging into every WSDL URL. likewise, you need to provide instructions to the test on how to validate the content returned by every WSDL URL, and also set an encoding format for each wsdl url. to enable administrators to easily configure the above per WSDL URL, eg enterprise provides a special interface. to access this interface, click on the encircled '+' button alongside the url text box in the test configuration page. alternatively, you can even click on the encircled '+' button adjacent to the WSDL URL parameter in the test configuration page. to know how to use this special interface, refer to section 6.2.7.1.

Parameters	Description
Operations	Once the WSDL URL(s) are specified, the operations that are offered by the web services and those that are to be monitored have to be configured. To select the required operations for monitoring, eG Enterprise provides a special interface. TO access this interface, click on the encircled '+' button alongside the Operations text box in the test configuration page. Alternatively, you can even click on the encircled '+' button adjacent to the OPERATIONS parameter in the test configuration page. To know how to use this special interface, refer to section 6.2.7.1.
Timeout	Specify the duration (in seconds) for which this test should wait for a response from the server. If there is no response from the server beyond the configured duration, the test will timeout. By default, this is set to 30 seconds.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the On option. To disable the capability, click on the Off option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
WSDL url availability:	Indicates whether the web service was able to respond successfully to the query made by the test.	Percent	Availability failures could be caused by several factors such as the web service process(es) being down, the web service being misconfigured, a network failure, etc. Temporary unavailability may also occur if the web service is overloaded. Availability is determined based on the response code returned by the service. A response code between 200 to 300 indicates that the service is available.

<table>
<tr><th>Measurement</th><th>Description</th><th>Measurement Unit</th><th>Interpretation</th></tr>
<tr><td>WSDL response time:</td><td>Indicates the time taken by the eG agent to get the configured web service.</td><td>Secs</td><td>Response time being high denotes a problem. Poor response times may be due to the service being overloaded or misconfigured. If the URL accessed involves the generation of dynamic content by the service, backend problems (e.g., an overload at the application server or a database failure) can also result in an increase in response time.</td></tr>
<tr><td>Port status:</td><td>Indicates whether/not the port of the web server is reachable.</td><td></td><td>The values reported by this measure and the corresponding numeric equivalents are listed in the table below:
<table><tr><th>Measure Values</th><th>Numeric Values</th></tr><tr><td>Yes</td><td>1</td></tr><tr><td>No</td><td>0</td></tr></table>
Note:
By default, this measure reports the above-mentioned Measure Values to indicate whether the server has been rebooted or not. In the graph of this measure however, the Measure Values are represented using the numeric equivalents only.</td></tr>
<tr><td>TCP connection availability:</td><td>Indicates whether the test managed to establish a TCP connection to the server.</td><td>Percent</td><td>Failure to establish a TCP connection may imply that either the web server process is not up, or that the process is not operating correctly. In some cases of extreme overload, the failure to establish a TCP connection may be a transient condition. As the load subsides, the server may start</td></tr>
</table>

Measurement	Description	Measurement Unit	Interpretation
			functioning properly again.
TCP connect time:	This measure quantifies the time for establishing a TCP connection to the web server host.	Secs	Typically, the TCP connection establishment must be very small (of the order of a few milliseconds). Since TCP connection establishment is handled at the OS-level, rather than by the application, an increase in this value signifies a system-level bottleneck on the host that supports the web server.
Server response time:	Indicates the time period between when the connection was established and when the web server sent back a response header to the client.	Secs	While the total response time may depend on several factors, this measure is typically, a very good indicator of a server bottleneck (e.g., because all the available server threads or processes are in use).
Response code:	The response code returned by the web server for the simulated request.	Number	A value between 200 and 300 indicates a good response. A 4xx value indicates a problem with the requested content (eg., page not found). A 5xx value indicates a server error.
Service availability:	Indicates whether/not the web service is available.	Percent	A value of 100 indicates that the web service is available and a value of 0 indicates that the web service is not available.
Operation status:	Indicates whether/not the configured operation is present in the web service.		This measure will not report metrics if the OPERATION parameter in the test configuration page is none in the test configuration page.
Operation Content length:	Indicates the response code returned by the server for the simulated request.	Number	A value between 200 and 300 indicates a good response. A 4xx value indicates a problem with the requested content (e.g., page not found). A 5xx value indicates a server error. This measure will not report metrics if

Measurement	Description	Measurement Unit	Interpretation
			the OPERATION parameter in the test configuration page is none or if an invalid Value is specified or if the Value is not specified in the HTML View tab while configuring the operation for monitoring in the test configuration page.
Operation Content validity:	This measure validates whether the operation was successful in executing the request made to it.	Percent	A value of 100% indicates that the content returned by the test is valid. A value of 0% indicates that the content may not be valid. This capability for content validation is especially important for multi-tier web applications. For example, a user may not be able to login to the web site but the server may reply back with a valid HTML page where in the error message, say, “Invalid Login” is reported. In this case, the availability will be 100 % (since we got a valid HTML response). If the test is configured such that the content parameter should exclude the string “Invalid Login”, in the above scenario content validity would have a value 0. This measure will not report metrics if the OPERATION parameter in the test configuration page is none or if an invalid Value is specified or if the Value is not specified in the HTML View tab while configuring the operation for monitoring in the test configuration page.
Operation execution time:	Indicates the time taken to invoke the configured operation in the web service.	Secs	This measure will not report metrics if the OPERATION parameter in the test configuration page is none or if an invalid Value is specified or if the Value is not specified in the HTML View tab

Measurement	Description	Measurement Unit	Interpretation
			while configuring the operation for monitoring in the test configuration page.

6.2.7.1 Configuring Multiple WSDL URLs for Monitoring

In order to enable the eG agent to connect to multiple WSDL URLs and pull out the required metrics from them, the eG administrative interface provides a special page using which different WSDL URLs and their corresponding operations that need to be monitored can be specified. To configure the WSDL URLs, do the following:

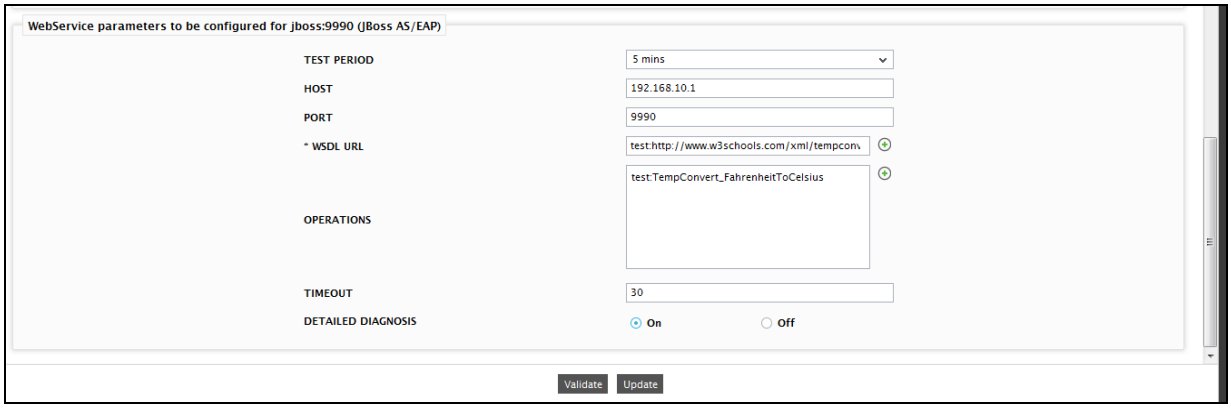

Figure 6.16: Configuring the WebService test

1. Click on the encircled '+' button alongside the **WSDL URL** text box. Figure 6.17 will then appear.

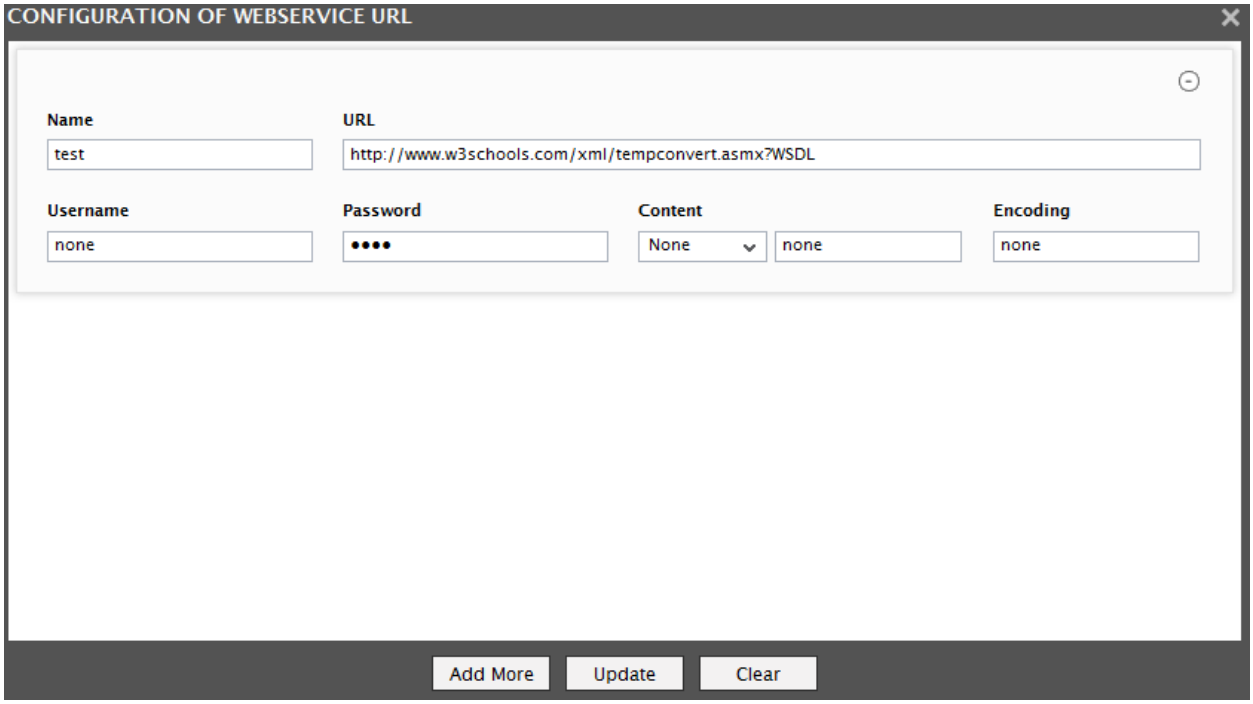

Figure 6.17: Figure 3.10: The WebService URL Configuration page

2. Specify the following in Figure 6.17:

- **Name:** Specify a unique name by which the WSDL URL you will be specifiying shortly will be referred to across the eG user interface. This is the name that will appear as the descriptor of this test.
- **URL:** Enter the WSDL URL of the web service that this test should access.
- **Username** and **Password:** These parameters are to be set only if a specific user name / password has to be specified to login to the web service (i.e., WSDL URL) that you have configured for monitoring. In this case, provide valid login credentials using the **Username** and **Password** text boxes. If the server on which **WebService** test executes supports 'Anonymous user access', then these parameters will take either of the following values:
  - A valid **Username** and **Password** for the configured **WSDL URL**
  - none in both the **Username** and **Password** text boxes of the configured WSDL URL, if no user authorization is required
  - Some servers however, support NTLM (Integrated Windows) authentication, where valid login credentials are mandatory. In other words, a none specification will not be supported

by such servers. Therefore, in this case, against each configured **WSDL URL**, you will have to provide a valid **Username** in the format: domainname\username, followed by a valid **Password**.

- Please be sure to check if your web service requires HTTP authentication while configuring this parameter. HTTP authentication typically involves a separate pop-up window when you try to access the page. Many services use HTTP POST for obtaining the user name and password and validating the user login. In such cases, the Username and Password have to be provided as part of the POST information and NOT as part of the **CREDENTIALS** specification for the WebService test.

- **Content**: The **Content** parameter has to be configured with an instruction:value pair that will be used to validate the content being returned by the test. If the **Content** value is None, no validation is performed. On the other hand, if you pick the Include option from the **Content** list, it indicates to the test that for the content returned by the web server to be valid, the content must include the specified value (a simple string search is done in this case). This value should be specified in the adjacent text box. Similarly, if the Exclude option is chosen from the **Content** drop-down, it indicates to the test that the server's output is valid if it does not contain the value specified in the adjacent text box. The Include or Exclude value you specify in the text box can include wildcard characters. For example, an Include instruction can be *Home page*.

- **Encoding**: Sometimes the eG agent has to parse the **WSDL URL** content with specific encoding other than the default (ISO-8859-1) encoding. In such a case, specify the type of encoding using which the eG agent can parse the **WSDL URL** content in the **Encoding** text box. By default, this value is none.

3. Similarly, you can add multiple URL specifications by clicking the **Add More** button. To remove a WSDL URL specification, click on the encircled '-' button corresponding to it. To clear all **WSDL URL** specifications, click the **Clear** button. To update all the changes you made, click the **Update** button.

4. Once **Update** is clicked, you will return to the test configuration page as shown in Figure 6.16. The **WSDL URL** text box in the test configuration page will display just the **Name**s - i.e., the unique display names - that you may have configured for the multiple WSDL URLs, as a comma-separated list. To view the complete WSDL URL specification, click the encircled '+' button alongside the **WSDL URL** text box, once again.

6.2.7.2 Configuring Multiple Operations for Monitoring - WebServiceTest

By default, the **WebServiceTest** test will be configured with the WSDL URLs that offer the web services that are to be monitored. To configure the operations that are offered by the WSDL URLs,

do the following:

1. Click on the encircled '+' button alongside the **OPERATIONS** text box as shown in Figure 6.16. Figure 6.18 will then appear.

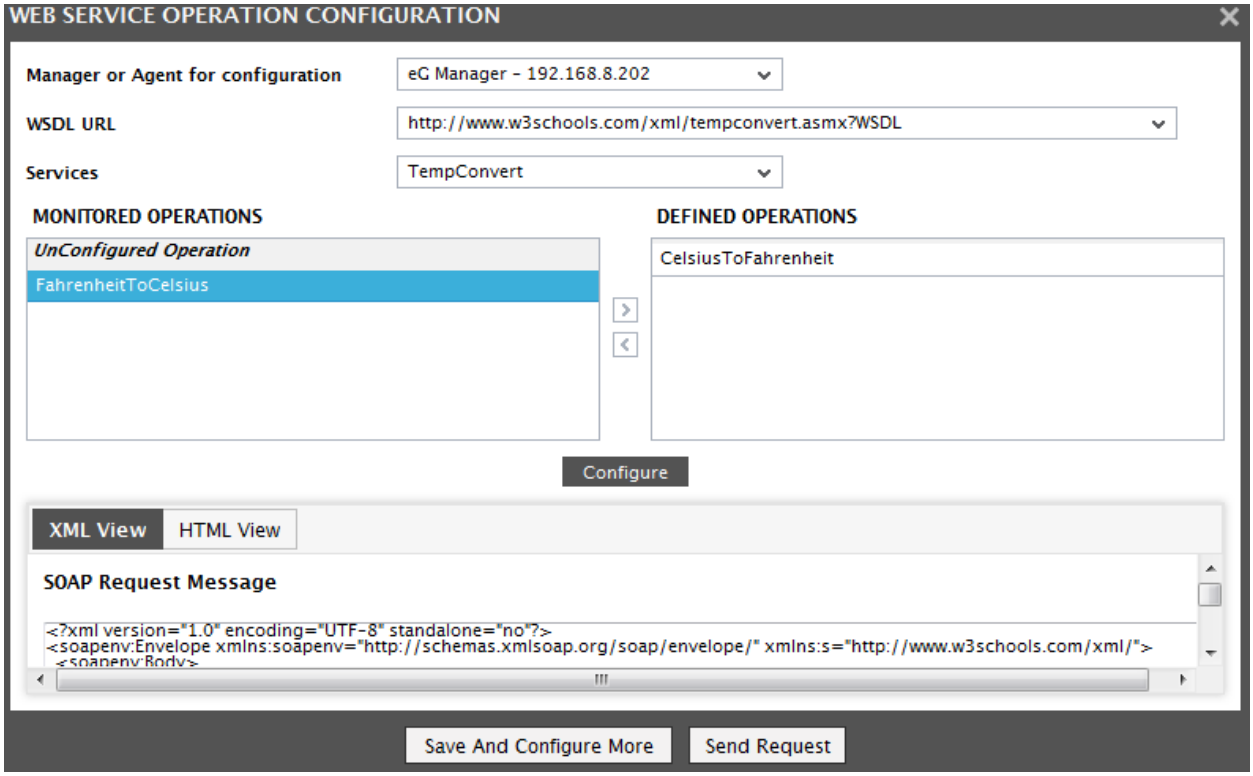

Figure 6.18: Figure 3.11: Configuring the Web Service Operation

2. Specify the following in Figure 6.18:

   - Manager/Agent for accessing WSDL URL: Select the eG agent or the eG Manager that is authorized to access the configured WSDL URL from this list.
   - WSDL URL: Once the eG agent/eG Manager is chosen from the Manager/Agent for accessing WSDL URL list, this list will be populated automatically with all the WSDL URLs specified in the **WSDL URL** text box (See Figure 6.16). Select the **WSDL URL** of your choice from this list.
   - Services: The web services offered by the chosen WSDL URL will then be populated in this list. Select a service of your choice from this list.

- The operations that are offered by the chosen service will then be populated in the **DEFINED OPERATIONS** list. To monitor a chosen operation, select the operation and click the < button. This will move the chosen operation to the **MONITORED OPERATIONS** list.
- Click the Configure button to save the changes.
- The eG agent uses SOAP requests to obtain the necessary metrics from the web service. Once the operation is configured, the XML View of the SOAP Request corresponding to the chosen operation will be generated and listed in the **XML View** tab. Likewise, the **HTML View** tab lists the **SOAP Parameter** that is passed to collect the required metrics for the chosen operation.
- To obtain operation-level statistics, it is important to specify a valid value in the VALUE text box of the HTML View tab as shown in Figure 6.18. Each time the test is executed, this value will be provided as an input to the chosen operation.

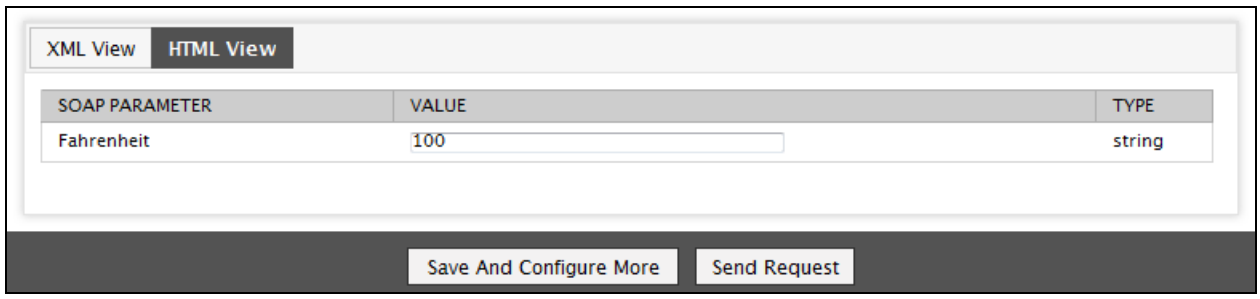

Figure 6.19: Figure 3.12: Specifying the value for the chosen operation

- Click the **Save and Cofigure More** button to save the changes made.
- If you wish to verify if the **VALUE** specified in the HTML View tab is valid, then you can do so by clicking the **Send Request** button. Figure 6.19 will then appear. If the value specified in the **VALUE** text box is indeed valid, then the operation will be performed on the value and the result will be specified. For example, if your chosen operation is FahrenheittoCelsius, the SOAP Parameter is Farenheit and the value that you wish to convert is 100, the result will be specified in the **WEB SERVICE RESPONSE** pop up window as below: <FahrenheitToCelsiusResult>37.7777777777778</FahrenheitToCelsiusResult>

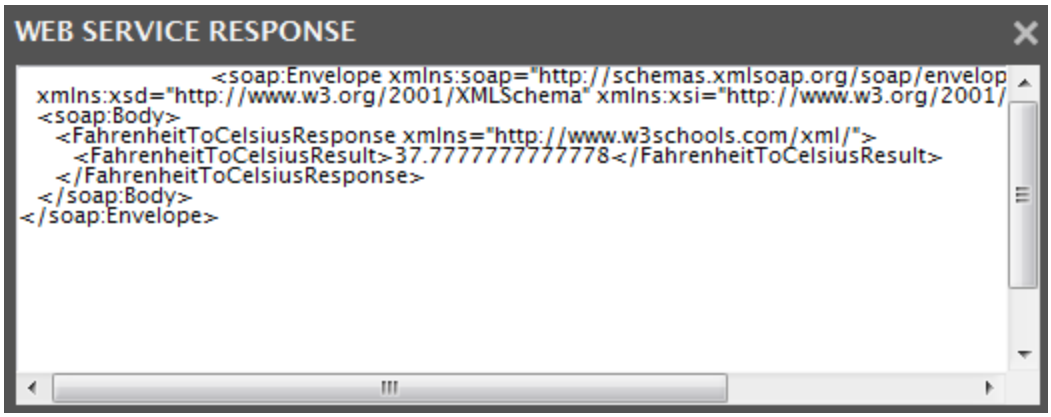

Figure 6.20: Figure 3.13: The value that appears when the operation is performed successfully

- If you have specified an invalid value, then a message as follows will be displayed in the pop up window:
  <FahrenheitToCelsiusResult>Error</FahrenheitToCelsiusResult>

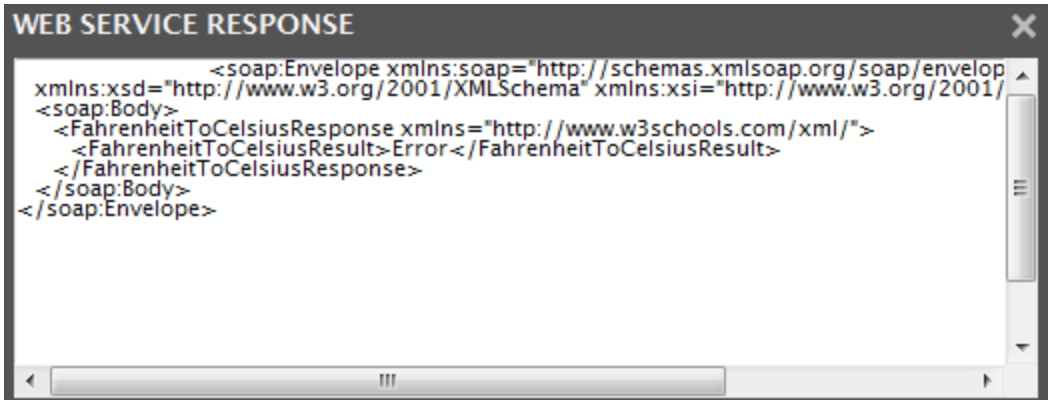

Figure 6.21: Figure 3.14: An Error appearing during value conversion

- If you do not specify a **VALUE** or specify an invalid value, operation-level statistics will not be collected by the eG agent and such metrics will not be available in the eG monitoring interface.

3. Similarly, you can configure multiple Operations by clicking the **Configure** button in Figure 6.18. To remove an operation, select the operation from the **MONITORED OPERATION** list and click the > button.
4. Once **Save and Configure More** button is clicked, you will return to the test configuration page (see Figure 6.18). The **OPERATIONS** text box in the test configuration page will display just the operations, as a comma-separated list. To view the complete operation specification, click the encircled '+' button alongside the **OPERATIONS** text box, once again.

6.3 The WAS Databases Layer

The **WAS Database** layer monitors the connection pools on the WebSphere application server.

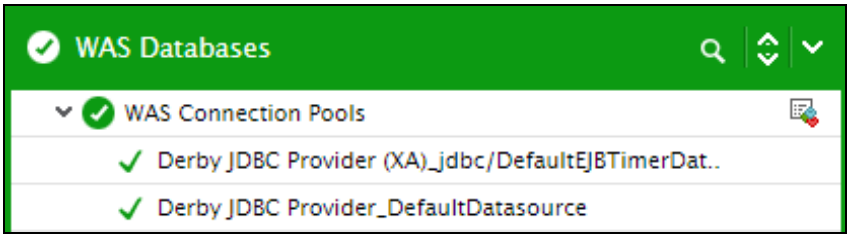

Figure 6.22: The test associated with the WAS Databases layer

6.3.1 WAS Connection Pools Test

This test monitors the usage of the connection pools on the WebSphere server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each connection pool being monitored

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
ServerHostName	Specify the host name of the application server instance being monitored.
AppPort	Specify the port number to be used for accessing the **egurkha** application that has been deployed on the server.
NodeName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within.

Parameters	Description
	• A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the **Node** that corresponds to the application server instance being monitored, and specify that name against the NodeName parameter.
ServerName	Provide the name of the server instance being monitored in the **SERVERNAME** text box. To know the server name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the Name of the monitored server instance, and specify that name against the ServerName parameter. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the host name that corresponds to the connector port of the *Deployment Manager* of the cluster as the ServerName. To determine the ServerName in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. Click on the **Details** button alongside the ports list. • If the SOAP port has been set as the connector port in your environment, then

Parameters	Description
	scroll down the page that appears next until you view the **Port Name**, **SOAP_CONNECTOR_ADDRESS**. Make note of the **Host** name that corresponds to this port name. If the RMI port is the connector port in your environment, then make note of the **Host** name that corresponds to the **Port** name, **BOOTSTRAP_ADDRESS**. • Specify this **Host** name as the ServerName.
ConnectorPort	The applications that are deployed on a server instance use the ConnectorPort for all internal communications with the application server. The connector port can be a SOAP port or an RMI port. The default connector port however, is the SOAP port. To know the connector port number, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on **Application Servers** within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. In the right pane, click on the server name link that corresponds to the server instance that is being monitored. • Doing so invokes the **Configuration** of the application server instance clicked on. Scroll down the **Configuration** tab page to view the **Communications** section. • Expand the **Ports** link in this section to view a list of ports. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the SOAP/RMI port of the *Deployment Manager* of the cluster as the ConnectorPort. To determine the ConnectorPort in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number**

Parameters	Description
	of the WebSphere server>\admin. • Login to the WebSphere Administrative console. • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ ADDRESS** as the ConnectorPort.
SSL	Select **Yes** if **SSL** (Secured Socket Layer) is to be used to connect to the WebSphere server, and **No** if it is not.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid **USER** name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the **USER** text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Allocate count	Indicates the number of connections allocated to	Number	

Measurement	Description	Measurement Unit	Interpretation
	the pool during the last measurement period.		
Close count	Indicates the number of connections released from the pool during the last measurement period.	Number	Ideally, this value should be low.
Create count	Indicates the number of connections created during the last measurement period.	Number	
Fault count	Indicates the number of connection timeouts in the pool in the last measurement period.	Number	Ideally, this value should be low. An unusually high value could indicate an application leak.
Freed count	Indicates the number of connections that were returned to the pool in the last measurement period	Number	A very low value of this measure could result in a shortage of connections in the pool.
Free pool size	Indicates the current number of free connections in the pool.	Number	
Pool size	Indicates the current size of the connection pool.	Number	
Jdbc time	Indicates the time taken by the JDBC queries to execute.	Secs	A low value is ideal.

6.4 The WAS Objects Layer

This layer transactions to the WebSphere server and reports how quickly the server processes the transactions. In addition, the test monitors the EJBs on the servers and reports the responsiveness of the bean methods.

Figure 6.23: The test mapped to the WAS Objects layer

6.4.1 WAS Transactions Test

Transactions are the key functionality of the WebSphere application server. The WASTransactions test monitors these transactions.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each transaction on the WebSphere application server

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
ServerHostName	Specify the host name of the application server instance being monitored.
AppPort	Specify the port number to be used for accessing the **egurkha** application that has been deployed on the server.
NodeName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the **Node** that corresponds to the application server instance being monitored, and

Parameters	Description
	specify that name against the NodeName parameter.
ServerName	Provide the name of the server instance being monitored in the **SERVERNAME** text box. To know the server name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the Name of the monitored server instance, and specify that name against the ServerName parameter. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the host name that corresponds to the connector port of the *Deployment Manager* of the cluster as the ServerName. To determine the ServerName in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. Click on the **Details** button alongside the ports list. • If the SOAP port has been set as the connector port in your environment, then scroll down the page that appears next until you view the **Port Name**, **SOAP_CONNECTOR_ADDRESS**. Make note of the **Host** name that corresponds to this port name. If the RMI port is the connector port in your environment, then

Parameters	Description
	make note of the **Host** name that corresponds to the **Port** name, **BOOTSTRAP_ADDRESS**. • Specify this **Host** name as the ServerName.
ConnectorPort	The applications that are deployed on a server instance use the ConnectorPort for all internal communications with the application server. The connector port can be a SOAP port or an RMI port. The default connector port however, is the SOAP port. To know the connector port number, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on **Application Servers** within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. In the right pane, click on the server name link that corresponds to the server instance that is being monitored. • Doing so invokes the **Configuration** of the application server instance clicked on. Scroll down the **Configuration** tab page to view the **Communications** section. • Expand the **Ports** link in this section to view a list of ports. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the SOAP/RMI port of the *Deployment Manager* of the cluster as the ConnectorPort. To determine the ConnectorPort in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console. • Expand the **System Administration** node in the tree-structure in the left pane of

Parameters	Description
	the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort.
SSL	Select **Yes** if **SSL** (Secured Socket Layer) is to be used to connect to the WebSphere server, and **No** if it is not.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid **USER** name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the **USER** text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Active count:	Indicates the number of concurrently active transactions.	Number	If the value of this measure is high, it signifies greater load on the server. This might increase the number of waiting transactions.
Commits count:	Indicates the number of transactions that were	Number	If the number of transactions that are being committed is very high, it

Measurement	Description	Measurement Unit	Interpretation
	committed since the last measurement period.		signifies load on the server. It might be caused when some locked transactions are released suddenly.
Rollback count:	Indicates the number of transactions that were rolled back since the last measurement period.	Number	A high value indicates a problem with the application or with some other (e.g. Database).
Begun count:	Indicates the number of transactions that were started on the server since the last measurement period.	Number	A high value indicates an overload on the server.
Timedout count:	Indicates the number of transactions that timed out since the last measurement period.	Number	A rise in the value could be due to the problem with the application or with some other dependent server like the database.
Transaction time:	Indicates the average duration of the transactions.	Secs	A high transaction duration value indicates increased load on the server.
Throughput:	Indicates the transaction throughput.	Reqs/Sec	

6.4.2 WAS Beans Test

The WAS Beans test automatically discovers the EJBs deployed on the WebSphere server and reports critical statistics pertaining to each of the EJBs.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each EJB deployed on the WebSphere application server

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
ServerHostName	Specify the host name of the application server instance being monitored.
AppPort	Specify the port number to be used for accessing the **egurkha** application that has been deployed on the server.
NodeName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the **Node** that corresponds to the application server instance being monitored, and specify that name against the NodeName parameter.
ServerName	Provide the name of the server instance being monitored in the **SERVERNAME** text box. To know the server name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the Name of the monitored server instance, and specify that name against the ServerName parameter. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the host name that corresponds to the connector port of the *Deployment Manager* of the cluster as the ServerName. To determine the ServerName in this case, do the following:

Parameters	Description
	• Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. Click on the **Details** button alongside the ports list. • If the SOAP port has been set as the connector port in your environment, then scroll down the page that appears next until you view the **Port Name**, **SOAP_CONNECTOR_ADDRESS**. Make note of the **Host** name that corresponds to this port name. If the RMI port is the connector port in your environment, then make note of the **Host** name that corresponds to the **Port** name, **BOOTSTRAP_ADDRESS**. • Specify this **Host** name as the ServerName.
ConnectorPort	The applications that are deployed on a server instance use the ConnectorPort for all internal communications with the application server. The connector port can be a SOAP port or an RMI port. The default connector port however, is the SOAP port. To know the connector port number, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on **Application Servers** within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. In the right pane, click on the server name link that corresponds to the server instance that is being monitored. • Doing so invokes the **Configuration** of the application server instance clicked

Parameters	Description
	on. Scroll down the **Configuration** tab page to view the **Communications** section. • Expand the **Ports** link in this section to view a list of ports. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the SOAP/RMI port of the *Deployment Manager* of the cluster as the ConnectorPort. To determine the ConnectorPort in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console. • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ ADDRESS** as the ConnectorPort.
SSL	Select **Yes** if **SSL** (Secured Socket Layer) is to be used to connect to the WebSphere server, and **No** if it is not.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid **USER** name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the **USER** text box should contain the default value 'none'.

Parameters	Description
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Create count	Indicates the number of times this EJB was created during the last measurement period.	Number	
Avg create time	Indicates the average time taken by a bean create call.	Secs	Ideally, this value should be low.
Remove count	Indicates the number of times this bean was removed during the last measurement period.	Number	
Avg remove time	Indicates the average time taken by a bean remove call.	Secs	
Activation count	Indicates the number of times this bean was activated during the last measurement period.	Number	
Avg activation time	Indicates the average time taken by a bean activate call, including the time at the database.	Secs	Ideally, this value should be low.
Store count	Indicates the number of times this bean was stored in the persistent storage during the last measurement period.	Number	
Avg store time	Indicates the average time for storing this bean in a	Secs	

Measurement	Description	Measurement Unit	Interpretation
	persistent storage.		
Instantiates count	Indicates the number of times this bean was instantiated during the last measurement period.	Number	A sudden increase in the number of instantiations indicates a bottleneck on the server. It may be due to greater load on the server or there might be a loophole in the application.
Freed count	Indicates the number of times during the last measurement period this bean object was freed.	Number	A very low value indicates a bottleneck on the server. This might affect the performance of the application.
Ready count	Indicates the number of concurrently ready beans.	Number	Greater the ready beans count, better will be the application performance.
Passive count	Indicates the number of beans in the passivated state during the last measurement.	Number	
Pooled count	Indicates the number of objects currently in the pool.	Number	
Drains count	Indicates the number of times during the last measurement period the daemon found the pool was idle and attempted to clean it.	Number	
Retrieve count:	Indicates the number of calls during the last measurement period retrieving an object from the pool.	Number	
Retrieve success count	Indicates the number of times a retrieve found an object in the pool, during the last measurement period.	Number	
Returns count	Indicates the number of calls returning an object to the pool, during the last measurement period.	Number	
Returns discard	Indicates the number of times	Number	Ideally, this value should be low. A

Measurement	Description	Measurement Unit	Interpretation
count	during the last measurement period the returning object was discarded because the pool was full.		consistent high value could indicate a full pool, and consequently, an overloaded server. You might then want to consider resizing the pool in order to accommodate more number of objects.
Methods call count	Indicates the number of method calls during the last measurement period.	Number	A high value indicates that the server is busy.
Methods response time	Indicates the average response time of the bean methods.	Secs	This value should be low for optimal performance of the application server. The value may go high, if there are more objects in the pool, which is a sign of overload.
Avg passivate time	The average time taken by a bean passivate call, including the time at the database	Secs	
Load count	The number of times this bean was loaded from the persistent storage during the last measurement period	Number	
Avg load time	The average time for loading a bean data from the persistent storage	Secs	Ideally, this value should be low.

6.5 The WAS Applications Layer

This layer extracts critical performance statistics pertaining to the following:

- The Http sessions on the server
- the web applications deployed on the WebSphere application server
- the synchronous and asynchronous requests received by WebSphere application server
- the Object Request Brokers (ORB)
- the web services mounted on a WebSphere server
- the JCA connection pools on the server

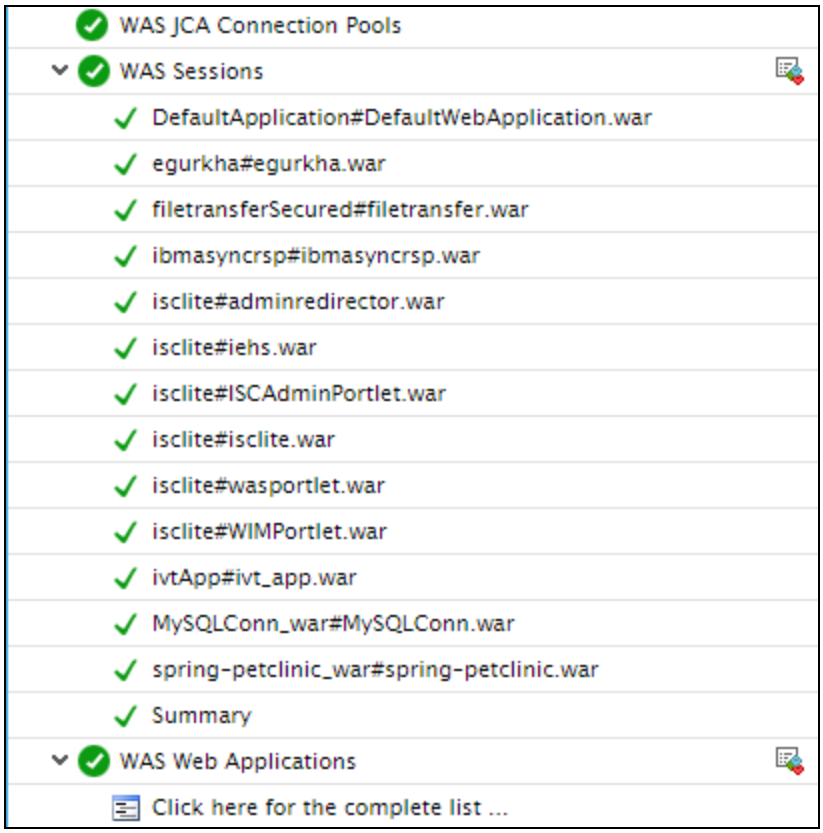

Figure 6.24: The tests associated with the WAS Applications layer

6.5.1 WAS Sessions Test

HTTP sessions form a part of any business application. This test monitors the HTTP sessions on the WebSphere application server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each session on the WebSphere application server

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server

Parameters	Description
ServerHostName	Specify the host name of the application server instance being monitored.
AppPort	Specify the port number to be used for accessing the **egurkha** application that has been deployed on the server.
NodeName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the **Node** that corresponds to the application server instance being monitored, and specify that name against the NodeName parameter.
ServerName	Provide the name of the server instance being monitored in the **SERVERNAME** text box. To know the server name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the Name of the monitored server instance, and specify that name against the ServerName parameter. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the host name that corresponds to the connector port of the *Deployment Manager* of the cluster as the ServerName. To determine the ServerName in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console

Parameters	Description
	• Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. Click on the **Details** button alongside the ports list. • If the SOAP port has been set as the connector port in your environment, then scroll down the page that appears next until you view the **Port Name**, **SOAP_CONNECTOR_ADDRESS**. Make note of the **Host** name that corresponds to this port name. If the RMI port is the connector port in your environment, then make note of the **Host** name that corresponds to the **Port** name, **BOOTSTRAP_ADDRESS**. • Specify this **Host** name as the ServerName.
ConnectorPort	The applications that are deployed on a server instance use the ConnectorPort for all internal communications with the application server. The connector port can be a SOAP port or an RMI port. The default connector port however, is the SOAP port. To know the connector port number, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on **Application Servers** within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. In the right pane, click on the server name link that corresponds to the server instance that is being monitored. • Doing so invokes the **Configuration** of the application server instance clicked on. Scroll down the **Configuration** tab page to view the **Communications** section. • Expand the **Ports** link in this section to view a list of ports. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been

Parameters	Description
	explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the SOAP/RMI port of the *Deployment Manager* of the cluster as the ConnectorPort. To determine the ConnectorPort in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console. • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort.
SSL	Select **Yes** if **SSL** (Secured Socket Layer) is to be used to connect to the WebSphere server, and **No** if it is not.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid **USER** name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the **USER** text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the

Parameters	Description
	Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Active count	Indicates the number of sessions that are currently accessed by requests.	Number	A high value indicates an increased workload on the server.
Live count	Indicates the number of sessions that are currently live.	Number	A high value indicates an increased workload on the server.
Create count	Indicates the number of sessions that were newly created since the last measurement period.	Number	A high value indicates a high level of activity in this application.
Invalidate count	Indicates the number of sessions that were invalidated since the last measurement period.	Number	This measure is indicative of the session usage pattern of this application.
Invalid session requests	Indicates the number of requests that were received since the last measurement period, for a session that no longer exists.	Number	
Cache discards	Indicates the number of session objects that were forced out of the cache.	Number	
New session failures	Indicates the number of times since the last measurement period, the request for a new session could not be handled because the value exceeded the maximum	Number	A consistent high value of this measure could indicate that many sessions have been alive for too long a time. A long session lifetime may occur due to one of the following reasons:

Measurement	Description	Measurement Unit	Interpretation
	session count.		• The application may not timeout the sessions after a fixed interval of time. • The sessions in the application might be active for a very long time.
Timeout invalidates	Indicates the number of sessions that were invalidated since the last measurement period, due to timeouts.	Number	

6.5.2 WAS Web Applications Test

This test reports statistics pertaining to the web applications deployed on a WebSphere server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each web application deployed on the WebSphere application server

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
ServerHostName	Specify the host name of the application server instance being monitored.
AppPort	Specify the port number to be used for accessing the **egurkha** application that has been deployed on the server.
NodeName	Specify the node name of the server instance being monitored. To know the node name, do the following:

Parameters	Description
	• Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the **Node** that corresponds to the application server instance being monitored, and specify that name against the NodeName parameter.
ServerName	Provide the name of the server instance being monitored in the **SERVERNAME** text box. To know the server name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the Name of the monitored server instance, and specify that name against the ServerName parameter. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the host name that corresponds to the connector port of the *Deployment Manager* of the cluster as the ServerName. To determine the ServerName in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the

Parameters	Description
	Configuration tab page, expand the **Ports** node. • A list of ports will then appear. Click on the **Details** button alongside the ports list. • If the SOAP port has been set as the connector port in your environment, then scroll down the page that appears next until you view the **Port Name**, **SOAP_CONNECTOR_ADDRESS**. Make note of the **Host** name that corresponds to this port name. If the RMI port is the connector port in your environment, then make note of the **Host** name that corresponds to the **Port** name, **BOOTSTRAP_ADDRESS**. • Specify this **Host** name as the ServerName.
ConnectorPort	The applications that are deployed on a server instance use the ConnectorPort for all internal communications with the application server. The connector port can be a SOAP port or an RMI port. The default connector port however, is the SOAP port. To know the connector port number, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on **Application Servers** within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. In the right pane, click on the server name link that corresponds to the server instance that is being monitored. • Doing so invokes the **Configuration** of the application server instance clicked on. Scroll down the **Configuration** tab page to view the **Communications** section. • Expand the **Ports** link in this section to view a list of ports. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the SOAP/RMI port of the *Deployment Manager* of the cluster as the ConnectorPort. To determine the ConnectorPort in this case, do the following:

Parameters	Description
	• Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console. • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ ADDRESS** as the ConnectorPort.
SSL	Select **Yes** if **SSL** (Secured Socket Layer) is to be used to connect to the WebSphere server, and **No** if it is not.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid **USER** name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the **USER** text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Loaded servlets	Indicates the number of servlets that were loaded since the last measurement period.	Number	
Reload count	Indicates the number of servlets that were reloaded since the last measurement period.	Number	
Request count	Indicates the number of requests that a servlet/JSP processed since the last measurement period.	Number	
Concurrent requests	Indicates the current number of requests processed concurrently.	Number	
Error count	Indicates the number of errors in JSP/Servlet since the last measurement period.	Number	
Service time	Indicates the response time of the JSP/servlet.	Secs	
Throughput	Indicates the throughput of this web application.	Reqs/Sec	

6.5.3 WAS Gateway Test

This test reports the number of synchronous and asynchronous requests received by and responses sent by the WebSphere server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every Gateway on the WebSphere application server being monitored

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
ServerHostName	Specify the host name of the application server instance being monitored.
AppPort	Specify the port number to be used for accessing the **egurkha** application that has been deployed on the server.
NodeName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the **Node** that corresponds to the application server instance being monitored, and specify that name against the NodeName parameter.
ServerName	Provide the name of the server instance being monitored in the **SERVERNAME** text box. To know the server name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the Name of the monitored server instance, and specify that name against the ServerName parameter. If the server instance being monitored is part of a WebSphere cluster, then you need

Parameters	Description
	to provide the host name that corresponds to the connector port of the *Deployment Manager* of the cluster as the ServerName. To determine the ServerName in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. Click on the **Details** button alongside the ports list. • If the SOAP port has been set as the connector port in your environment, then scroll down the page that appears next until you view the **Port Name**, **SOAP_CONNECTOR_ADDRESS**. Make note of the **Host** name that corresponds to this port name. If the RMI port is the connector port in your environment, then make note of the **Host** name that corresponds to the **Port** name, **BOOTSTRAP_ADDRESS**. • Specify this **Host** name as the ServerName.
ConnectorPort	The applications that are deployed on a server instance use the ConnectorPort for all internal communications with the application server. The connector port can be a SOAP port or an RMI port. The default connector port however, is the SOAP port. To know the connector port number, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on **Application Servers** within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. In the right pane, click on the server

Parameters	Description
	name link that corresponds to the server instance that is being monitored. • Doing so invokes the **Configuration** of the application server instance clicked on. Scroll down the **Configuration** tab page to view the **Communications** section. • Expand the **Ports** link in this section to view a list of ports. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the SOAP/RMI port of the *Deployment Manager* of the cluster as the ConnectorPort. To determine the ConnectorPort in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console. • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ ADDRESS** as the ConnectorPort.
SSL	Select **Yes** if **SSL** (Secured Socket Layer) is to be used to connect to the WebSphere server, and **No** if it is not.
User	If security has been enabled for the WebSphere server being monitored, then provide

Parameters	Description
	a valid **USER** name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the **USER** text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Sync request count	Indicates the number of synchronous requests that were made, since the last measurement period.	Number	
Sync response count	Indicates the number of synchronous responses that were sent, since the last measurement period.	Number	
Async request count	Indicates the number of asynchronous requests that were made, since the last measurement period.	Number	
Async response count	Indicates the number of asynchronous responses that were made, since the last measurement period.	Number	

6.5.4 WAS ORB Performance Test

This test reports statistics pertaining to the ORBs (Object Request Broker) on a WebSphere server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each ORB being monitored

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
ServerHostName	Specify the host name of the application server instance being monitored.
AppPort	Specify the port number to be used for accessing the **egurkha** application that has been deployed on the server.
NodeName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the **Node** that corresponds to the application server instance being monitored, and specify that name against the NodeName parameter.
ServerName	Provide the name of the server instance being monitored in the **SERVERNAME** text box. To know the server name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the Name of the monitored server instance, and specify that name against the

Parameters	Description
	ServerName parameter. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the host name that corresponds to the connector port of the *Deployment Manager* of the cluster as the ServerName. To determine the ServerName in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. Click on the **Details** button alongside the ports list. • If the SOAP port has been set as the connector port in your environment, then scroll down the page that appears next until you view the **Port Name**, **SOAP_CONNECTOR_ADDRESS**. Make note of the **Host** name that corresponds to this port name. If the RMI port is the connector port in your environment, then make note of the **Host** name that corresponds to the **Port** name, **BOOTSTRAP_ADDRESS**. • Specify this **Host** name as the ServerName.
ConnectorPort	The applications that are deployed on a server instance use the ConnectorPort for all internal communications with the application server. The connector port can be a SOAP port or an RMI port. The default connector port however, is the SOAP port. To know the connector port number, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on **Application Servers** within.

Parameters	Description
	• A list of application server instances and their corresponding node names will then appear in the right pane of the console. In the right pane, click on the server name link that corresponds to the server instance that is being monitored. • Doing so invokes the **Configuration** of the application server instance clicked on. Scroll down the **Configuration** tab page to view the **Communications** section. • Expand the **Ports** link in this section to view a list of ports. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the SOAP/RMI port of the *Deployment Manager* of the cluster as the ConnectorPort. To determine the ConnectorPort in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console. • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ ADDRESS** as the ConnectorPort.
SSL	Select **Yes** if **SSL** (Secured Socket Layer) is to be used to connect to the

Parameters	Description
	WebSphere server, and **No** if it is not.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid **USER** name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the **USER** text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Requests count	Indicates the number of requests sent to the ORB, since the last measurement period.	Number	
Concurrent requests count	Indicates the current number of requests concurrently processed by the ORB.	Number	
Lookup time	Indicates the amount of time taken to lookup an object reference before method dispatch can be carried out.	Secs	An excessively long time may indicate an EJB container lookup problem.
Processing time	Indicates the time that it takes a registered portable interceptor to run.	Secs	

6.5.5 WAS Web Service Test

This test reports statistics pertaining to the web services mounted on a WebSphere server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for every web service on the WebSphere application server being monitored

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
ServerHostName	Specify the host name of the application server instance being monitored.
AppPort	Specify the port number to be used for accessing the **egurkha** application that has been deployed on the server.
NodeName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the **Node** that corresponds to the application server instance being monitored, and specify that name against the NodeName parameter.
ServerName	Provide the name of the server instance being monitored in the **SERVERNAME** text box. To know the server name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the Name of the monitored server instance, and specify that name against the

Parameters	Description
	ServerName parameter. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the host name that corresponds to the connector port of the *Deployment Manager* of the cluster as the ServerName. To determine the ServerName in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. Click on the **Details** button alongside the ports list. • If the SOAP port has been set as the connector port in your environment, then scroll down the page that appears next until you view the **Port Name**, **SOAP_CONNECTOR_ADDRESS**. Make note of the **Host** name that corresponds to this port name. If the RMI port is the connector port in your environment, then make note of the **Host** name that corresponds to the **Port** name, **BOOTSTRAP_ADDRESS**. • Specify this **Host** name as the ServerName.
ConnectorPort	The applications that are deployed on a server instance use the ConnectorPort for all internal communications with the application server. The connector port can be a SOAP port or an RMI port. The default connector port however, is the SOAP port. To know the connector port number, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on **Application Servers** within.

Parameters	Description
	• A list of application server instances and their corresponding node names will then appear in the right pane of the console. In the right pane, click on the server name link that corresponds to the server instance that is being monitored. • Doing so invokes the **Configuration** of the application server instance clicked on. Scroll down the **Configuration** tab page to view the **Communications** section. • Expand the **Ports** link in this section to view a list of ports. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the SOAP/RMI port of the *Deployment Manager* of the cluster as the ConnectorPort. To determine the ConnectorPort in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console. • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ ADDRESS** as the ConnectorPort.
SSL	Select **Yes** if **SSL** (Secured Socket Layer) is to be used to connect to the

Parameters	Description
	WebSphere server, and **No** if it is not.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid **USER** name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the **USER** text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Loaded services	Indicates the number of web services loaded by the application server since the last measurement period.	Number	
Dispatched requests	Indicates the number of requests that were dispatched by the service to a target code, since the last measurement period.	Number	
Processed requests	Indicates the number of requests dispatched successfully with corresponding replies, since the last measurement period.	Number	
Received requests	Indicates number of requests received by the service.	Number	
Response time	Indicates the average time between the receipt of a request and the return of a	Secs	A very high response time could indicate a bottleneck at the web service.

Measurement	Description	Measurement Unit	Interpretation
	reply.		
Request response time	Indicates the average time between the receipt of the request and the dispatch for processing the request.	Secs	
Reply response time	Indicates the average time between the dispatch of the reply and return of the reply.	Secs	

6.5.6 WAS JCA Connection Pools Test

This test monitors the usage of the JCA connection pools on the WebSphere Application server.

Target of the test : A WebSphere application server

Agent deploying the test : An internal agent

Outputs of the test : One set of results for each connection pool being monitored

Configurable parameters for the test

Parameters	Description
Test period	How often should the test be executed .
Host	The IP address of the WebSphere application server
Port	The port number of the WebSphere application server
ServerHostName	Specify the host name of the application server instance being monitored.
AppPort	Specify the port number to be used for accessing the **egurkha** application that has been deployed on the server.
NodeName	Specify the node name of the server instance being monitored. To know the node name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within.

Parameters	Description
	• A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the **Node** that corresponds to the application server instance being monitored, and specify that name against the NodeName parameter.
ServerName	Provide the name of the server instance being monitored in the **SERVERNAME** text box. To know the server name, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on the **Application Servers** link within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. From this list, you can figure out the Name of the monitored server instance, and specify that name against the ServerName parameter. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the host name that corresponds to the connector port of the *Deployment Manager* of the cluster as the ServerName. To determine the ServerName in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number of the WebSphere server>\admin**. • Login to the WebSphere Administrative console • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. Click on the **Details** button alongside the ports list. • If the SOAP port has been set as the connector port in your environment, then

Parameters	Description
	scroll down the page that appears next until you view the **Port Name**, **SOAP_CONNECTOR_ADDRESS**. Make note of the **Host** name that corresponds to this port name. If the RMI port is the connector port in your environment, then make note of the **Host** name that corresponds to the **Port** name, **BOOTSTRAP_ADDRESS**. • Specify this **Host** name as the ServerName.
ConnectorPort	The applications that are deployed on a server instance use the ConnectorPort for all internal communications with the application server. The connector port can be a SOAP port or an RMI port. The default connector port however, is the SOAP port. To know the connector port number, do the following: • Login to the WebSphere Administrative Console. • Expand the **Servers** node in the tree structure in the left pane of the console, and click on **Application Servers** within. • A list of application server instances and their corresponding node names will then appear in the right pane of the console. In the right pane, click on the server name link that corresponds to the server instance that is being monitored. • Doing so invokes the **Configuration** of the application server instance clicked on. Scroll down the **Configuration** tab page to view the **Communications** section. • Expand the **Ports** link in this section to view a list of ports. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ADDRESS** as the ConnectorPort. If the server instance being monitored is part of a WebSphere cluster, then you need to provide the SOAP/RMI port of the *Deployment Manager* of the cluster as the ConnectorPort. To determine the ConnectorPort in this case, do the following: • Connect to the WebSphere Administrative console using the URL: **http://<IP address of the WebSphere server:Port number of the WebSphere server>\adminor https://<IP address of the WebSphere server:Port number**

Parameters	Description
	of the WebSphere server>\admin. • Login to the WebSphere Administrative console. • Expand the **System Administration** node in the tree-structure in the left pane of the console and click on the **Deployment Manager** sub-node within. • In the right panel, click on the **Configuration** tab page to view the configuration of the **Deployment Manager**. In the **Additional Properties** section of the **Configuration** tab page, expand the **Ports** node. • A list of ports will then appear. If the default connector port is in use, then the port number displayed against **SOAP_CONNECTOR_ADDRESS** should be specified as the ConnectorPort. If an RMI port has been explicitly set as the connector port, then specify the port number displayed against **BOOTSTRAP_ ADDRESS** as the ConnectorPort.
SSL	Select **Yes** if **SSL** (Secured Socket Layer) is to be used to connect to the WebSphere server, and **No** if it is not.
User	If security has been enabled for the WebSphere server being monitored, then provide a valid **USER** name to login to the WebSphere server. If the WebSphere server does not require any authentication, then the **USER** text box should contain the default value 'none'.
Password	If security has been enabled for the WebSphere server being monitored, then provide the Password that corresponds to the specified User name. If the WebSphere server does not require any authentication, then leave the Password text box with its default setting.
Confirm Password	If security has been enabled, confirm the specified Password by retyping it in the Confirm Password text box. If the WebSphere server does not require any authentication, then leave the Confirm Password text box with its default setting.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Connections allocated	Indicates the number of connections allocated to	Number	

Measurement	Description	Measurement Unit	Interpretation
	the pool during the last measurement period.		
Connections freed	Indicates the number of connections that were returned to the pool in the last measurement period.	Number	A very low value of this measure could result in a shortage of connections in the pool.
Connections created	Indicates the number of connections created during the last measurement period.	Number	
Connections closed	Indicates the number of connections released from the pool during the last measurement period.	Number	Ideally, this value should be low.
Connections in pool	Indicates the number of connections in this connection pool.	Number	
Free connections in pool	Indicates the number of free connection available in this connection pool.	Number	
Waiting threads	Indicates the number of threads that are currently waiting in this connection pool	Number	
Faults	Indicates the number of faults (for e.g., timeouts) that occurred in this connection pool per second during the last measurement period.	Faults/sec	Ideally, the value of this measure should be zero.
Percent of the pool in use	Indicates the current utilization of this connection pool in percentage.	Percent	This value of this measure is based on the number of connections that are configured for this connection pool.
Percent of the time	Indicates the number of	Percent	

Measurement	Description	Measurement Unit	Interpretation
that all connections are in use	times (expressed as percentage) all the connections of this connection pool were in use.		
Avg. use time of connections:	Indicates the average time for which the connections of this connection pool were in use.	MilliSec	
Avg. wait time to grand connection	Indicates the average time a client has to wait before the connections were granted from this connection pool.	MilliSec	
Managed connections	Indicates the number of managed connections that are currently in use in this connection pool.	Number	
Connections associated with physical connection	Indicates the number of connections that are associated with physical connections in this connection pool.	Number	

6.6 The Java Transactions Layer

By default, this layer will not be available for the IBM WebSphere application server. This is because, the **Java Business Transactions** test mapped to this layer is disabled by default. To enable the test, follow the *Agents -> Tests -> Enable/Disable* menu sequence, select *IBM WebSphere* as the **Component type**, *Performance* as the **Test type**, and then select **Java Business Transactions** from the **DISABLED TESTS** list. Click the **Enable** button to enable the selected test, and click the **Update** button to save the changes.

6.6.1 Java Business Transactions Test

The responsiveness of a transaction is the key determinant of user experience with that transaction; if response time increases, user experience deteriorates. To make users happy, a Java business

transaction should be rapidly processed by each of the JVM nodes in its path. Processing bottlenecks on a single JVM node can slowdown/stall an entire business transaction or can cause serious transaction errors. This in turn can badly scar the experience of users. To avoid this, administrators should promptly identify slow/stalled/errored transactions, isolate the JVM node on which the slowness/error occurred, and uncover what caused the aberration on that node – is it owing to SQL queries executed by the node? Or is it because of external calls – eg., async calls, SAP JCO calls, HTTP calls, etc. - made by that node? The **Java Business Transactions** test helps with this!

This test runs on a BTM-enabled JVM in an IT infrastructure, tracks all the transaction requests received by that JVM, and groups requests based on user-configured pattern specifications. For each transaction pattern, the test then computes and reports the average time taken by that JVM node to respond to the transaction requests of that pattern. In the process, the test identifies the slow/stalled transactions of that pattern, and reports the count of such transactions and their responsiveness. Detailed diagnostics provided by the test accurately pinpoint the exact transaction URLs that are slow/stalled, the total round-trip time of each transaction, and also indicate when such transaction requests were received by that node. The slowest transaction in the group can thus be identified.

For this test to run and report metrics on a WebSphere server, you first need to BTM-enable the WebSphere JVM. To know how, refer to the Installing eG Java BTM on an IBM WebSphere topic in the *Java Business Transaction Monitoring* document.

Then, proceed to configure this test. Refer to the Java Business Transactions Test topic in the *Java Business Transaction Monitoring* document to know how to configure this test and learn about the metrics it reports.

Chapter 7: Troubleshooting

If any of the WebSphere tests is not reporting measures, then try to connect to the URL of the following format:

http://<WebSphereIP>:<WebSpherePort>/egurkha/egurkha/EgWebSphere.jsp?module=<moduleName>&hostname=<serverHostName>&&nodename=<nodeName>&connectorport=<SOAPConnectorPort>&server=<serverName>&user=<userName>&password=<password>

Here, specify

hostIP = IP of the machine in which the WebSphere Server is running.

hostPort = The port on which the server is running.

moduleName = The name of the module for which measures are required. This will vary from one test to another.

serverHostName = The value of the **SERVERHOSTNAME** parameter of the test

SOAPConnectorPort = The SOAP connector port that is mentioned against the **CONNECTORPORT** parameter of the test

userName and password = If security is enabled for the WebSphere server being monitored, then provide a valid user name and password

server = The value of the **SERVERNAME** parameter of the test

nodeName = The value of the **NODENAME** parameter of the test

Examples:

Note:

All the examples provided below pertain to WebSphere environments, where security is not enabled.

1. If the WAS Oraclen test is not reporting measures, then use the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere?module=OraclenModule&hostname=egitlab04&nodename=egurkha07Node01&connectorport=8880&server=server1

2. If the WAS Connection Pools test is not reporting measures, then use the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere?module=connectionPoolModule&
   hostname=egitlab04&nodename=egurkha07Node01&connectorport=8880&server=server1

3. If the WAS Cache test is not reporting measures, then use the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere?module=cacheModule&
   hostname=egitlab04&nodename=egurkha07Node01&connectorport=8880&server=server1

4. If the WAS Object Pools test is not reporting measures, then use the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere?module=objectPoolModule&
   hostname=egitlab04&nodename=egurkha07Node01&connectorport=8880&server=server1

5. If the WAS Gateway test is not reporting measures, then use the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere?module=wsgwModule&
   hostname=egitlab04&nodename=egurkha07Node01&connectorport=8880&server=server1

6. If the WAS ORB Performance test is not reporting measures, then use the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere?module=orbPerfModule&
   hostname=egitlab04&nodename=egurkha07Node01&connectorport=8880&server=server1

7. If the WAS Web Applications test is not reporting measures, then use the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere?module=webAppModule&
   hostname=egitlab04&nodename=egurkha07Node01&connectorport=8880&server=server1

8. If the WAS Web Service test is not reporting measures, then use the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere?module=webServicesModule&
   hostname=egitlab04&nodename=egurkha07Node01&connectorport=8880&server=server1

9. If the WAS Sessions test is not reporting measures, then use the following URL:

   http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere?module=servletSessionsModule&
   hostname=egitlab04&nodename=egurkha07Node01&connectorport=8880&server=server1

10. If the WAS Transactions test is not reporting measures, then use the following URL:

    http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere?module=transactionModule&
    hostname=egitlab04&nodename=egurkha07Node01&connectorport=8880&server=server1

11. If the WAS JVM test is not reporting measures, then use the following URL:

    http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere?module=jvmRuntimeModule&
    hostname=egitlab04&nodename=egurkha07Node01&connectorport=8880&server=server1

12. If the WAS Threads test is not reporting measures, then use the following URL:

    http://192.168.10.77:9080/egurkha/egurkha/EgWebSphere?module=threadPoolModule&
    hostname=egitlab04&nodename=egurkha07Node01&connectorport=8880&server=server1

About eG Innovations

eG Innovations provides intelligent performance management solutions that automate and dramatically accelerate the discovery, diagnosis, and resolution of IT performance issues in on-premises, cloud and hybrid environments. Where traditional monitoring tools often fail to provide insight into the performance drivers of business services and user experience, eG Innovations provides total performance visibility across every layer and every tier of the IT infrastructure that supports the business service chain. From desktops to applications, from servers to network and storage, from virtualization to cloud, eG Innovations helps companies proactively discover, instantly diagnose, and rapidly resolve even the most challenging performance and user experience issues.

eG Innovations is dedicated to helping businesses across the globe transform IT service delivery into a competitive advantage and a center for productivity, growth and profit. Many of the world's largest businesses use eG Enterprise to enhance IT service performance, increase operational efficiency, ensure IT effectiveness and deliver on the ROI promise of transformational IT investments across physical, virtual and cloud environments.

To learn more visit www.eginnovations.com.

Contact Us

For support queries, email support@eginnovations.com.

To contact eG Innovations sales team, email sales@eginnovations.com.