Estimated time 1:00

WebSphere Virtual Enterprise: **Dynamic operations for non-WebSphere endpoints**

Non WebSphere dynamic operations

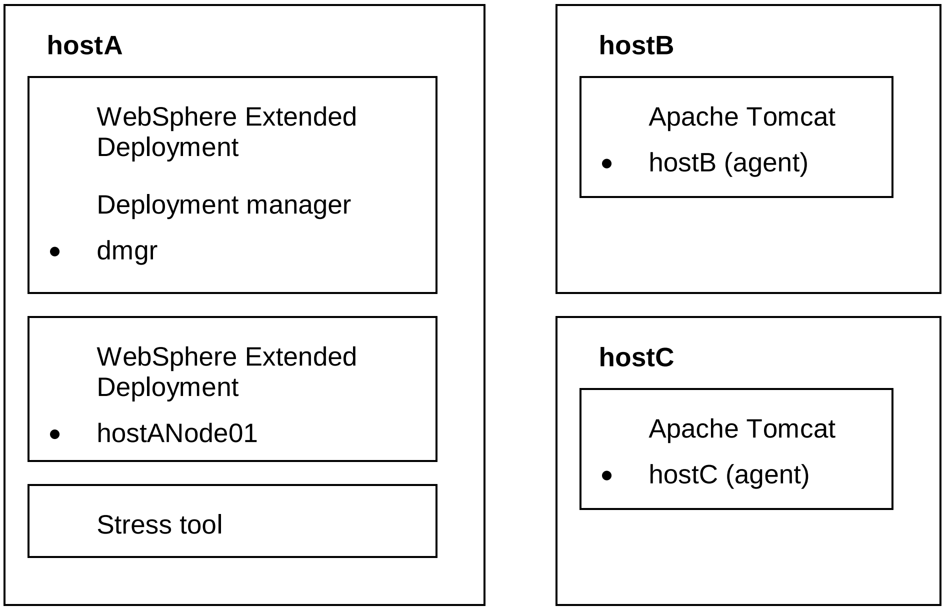
What this exercise is about

The objective of this lab is to provide you with an understanding of how to configure WebSphere Extended Deployment for dynamic operations to support non-WebSphere endpoints, specifically using Apache Tomcat servers as a representative example of a non-WebSphere endpoint.

Lab requirements

This lab assumes that this setup is complete before starting the lab. If you do not have the WebSphere portions of this environment set up, first complete the Installation Lab exercise, and then perform the steps specified in Part 1 of this exercise. The Apache Tomcat servers are already installed on hostB and hostC, and the MicroWebApp has been installed on the Apache Tomcat servers.

Non WebSphere dynamic operations



- The lab requires three machines: hostA, hostB, and hostC
- Deployment manager, on-demand router node, ODR and the stress tool are installed on hostA.
- HostB and HostC each contain a stand-alone Apache Tomcat V5.5 server installation with the MicroWebApp application already installed on each of the two Apache Tomcat servers.
- HostB and HostC each contain the WebSphere agent binaries installed.
- In a production environment the deployment manager should not be on the same node as the on-demand router. It will normally be on a machine by itself, though it could be co-located with an application server node.

What you should be able to do

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

- Configure the WebSphere agent, Tomcat dynamic cluster, service policy, transaction class
- Install a representation of the Tomcat application in WebSphere
- Configure WebSphere Extended Deployment so that:
    - You can start and stop the Apache Tomcat servers from the WebSphere administrative console
    - The on-demand router (ODR) can route http traffic to the Tomcat servers that are active
    - Placement can start another instance of the Tomcat server on another node if the processor utilization exceeds 90 percent
    - Dynamic operations will initiate tasks for excessive processor utilization or service policy violations
- Review statistical information about the Tomcat server traffic by studying the visualization charts in the administrative console

Introduction

Application server placement and dynamic operations are powerful features of WebSphere Extended Deployment. Extended Deployment allows applications to be virtually present in a ‘cluster’, much like the virtual memory in an operating system. In WebSphere Extended Deployment V6.1, application server placement and dynamic operations have been extended so that these facilities can facilitate routing http traffic to “middleware” application server types and start these servers as needed by your operational policy setting. An agent on each node allows autonomic control of the servers and some manual control of these servers using the administrative console. The agent supplies statistical information to the dynamic operations and application server placement facilities to assist in health and service policy enforcement. The agents appear in the administrative console as unmanaged nodes.

Once the agents and the application servers are defined to WebSphere and once the representation of their associated applications is created within WebSphere, the placement of applications onto those servers can be determined by operational policies, and the work will be routed to the application by an intelligent workload manager. Operational policy consists of two main classes: service policies and transaction classes, which are used to categorize work. The work, before being run, has its operation policy read by the on-demand router in order to determine the proper flow of work for completion, given the available server resource. Service policies are the main building blocks of the operational policy.

Visualization in Extended Deployment allows you to see a visual representation of the operations occurring for these servers, much like it does for WebSphere Application servers. You can drill down to more specific views from the operation center on your desktop. By configuring filters you can create customized charting, and navigate your chart views depending on your preference settings.

In this lab, a sample application is already installed on a Tomcat server on hostB and hostC nodes. You will install a WebSphere agent on hostB and hostC. You will use the WebSphere administrative console to define a dynamic cluster for these two server nodes. Through the administrative console, you will create a representation of the application within WebSphere Extended Deployment so that Extended Deployment can route to it and monitor the servers. You will define an operational policy for this environment and verify that application server placement starts a server instance when it is required. The mapping that you will create between the Web modules, dynamic clusters, service policies and transaction classes is shown in table here:

Dynamic cluster	Web module	Transaction class	Service policy
Tomcat_DC	MicroWebApp	Tomcat_TC	Tomcat_SP

The goal that you will create for the service policy is shown below.

Service policy	Goal	Importance
Tomcat_SP	500ms	Very high

Exercise instructions

This exercise assumes you have created a deployment manager and the ODR server. Instructions are provided to create the ODR server within hostANode01, but you can skip that portion of the lab is your ODR server is already created. It assumes you have installed the WebSphere agent software binaries on hostB and hostC, collocated with the Apache Tomcat installations. Finally, you must also have installed the MicroWebApp application (or your own test application) within Apache Tomcat installations.

If your environment is different from that described above, you may adjust the instructions to match your installation (for example, host, node names, applications, and so on).

Some instructions in this lab are Windows® operating-system specific. If you plan on running the lab on an operating-system other than Windows, you will need to run the appropriate commands, and use appropriate files (.sh or .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references, as follows:

Reference variable	Windows location	AIX® or UNIX® location
<WAS_HOME>	C:\WebSphere\AppServer	/usr/WebSphere/AppServer /opt/WebSphere/AppServer
<XDAgent>	C:\WebSphere\XDAgent	/usr/WebSphere/XDAgent /opt/WebSphere/XDAgent
<LAB_FILES>	C:\LabFilesXD	/tmp/LabFilesXD
<LAB_NAME>	PlacementLab	PlacementLab

Note for Windows users: When directory locations are passed as parameters to a Java program such as EJBdeploy or wsadmin, it is necessary to replace the backslashes with forward slashes to follow the Java convention. For example, replace C:\LabFilesXD\ with C:/LabFilesXD/

Part 1: Create an on-demand router

The “Lab requirements” section of this lab shows a multi-machine environment that is required to complete the exercise. This section will walk you through creating such an environment, provided that you have already completed the installation lab exercise.

____ 1. Start the deployment manager.

__ a. On **hostA**, open a command prompt.

__ b. Change directories to **C:\WebSphere\AppServer\profiles\dmgr\bin.**

__ c. Enter this command to start the deployment manager: **startmanager.**

__ d. Wait for the deployment manager to start. Verify that this line appears in the command prompt window.

```
ADMU3000I: Server dmgr open for e-business; process id is XXXX
```

____ 2. Start the hostANode01’s node agent.

__ a. Change directories to **C:\WebSphere\AppServer\profiles\hostANode01\bin.**

__ b. Enter this command to start the node agent on the node: **startNode**

____ 3. WebSphere Extended Deployment provides a script (createodr.jacl) that automatically creates an on-demand router named ‘odr’. Run it to create an on-demand router for this lab exercise.

__ a. On **hostA**, Open a command prompt.

__ b. Change directories to **C:\WebSphere\AppServer\bin.**

__ c. Enter this command to create an on-demand router server on node hostANode01:

wsadmin –f createodr.jacl hostANode01

Successful creation will be indicated by these final key final messages:

```
...
createodr: creating a server ODR  .....
createodr: saving the configuration
createodr: invoking synchronization for node...
createodr: Done with synchronization
createodr: done.
```

Note: This script will create an on-demand router that listens for incoming HTTP requests on port 80. Ensure that your system is not running a Web server or other process that is also listening on port 80.

If a server named odr already exists on this node, you will see the message:

```
createodr: Error—Server odr already configured on node hostANode01
```

Part 2: Start the server processes

You can start the remote node agents using the procedures below, or by using the procedures described in Appendix A – Starting a middleware node from the administrative console on page 40.

____ 4. Start the hostANode01 node agent, if it is not already running.

__ a. On **hostA**, change directories to **C:\WebSphere\AppServer\profiles\hostANode01\bin.**

__ b. Enter this command to start the node agent on the hostANode01 node: **startNode**

____ 5. Start the on-demand router (ODR) server.

__ a. On **hostA**, change directories to **C:\WebSphere\AppServer\profiles\hostANode01\bin.**

__ b. Enter this command to start the on-demand router on the node: **startServer ODR**

Non WAS Dynamic Operations

Part 3: Create, federate and start the agents

Note: The WebSphere agent software is installed on hostB and hostC in the **C:\WebSphere\XDAgent** directory.

____ 6. On hostB, navigate to **C:\WebSphere\XDAgent\bin** within a command window.

____ 7. Issue this command to federate the node:

addAgent –host hostA –port 9060

The following indicates a successful federation of the agent. The agent will have the same name as the local host setting.

```
C:\WebSphere\XDAgent\bin>addAgent -host hostA -port 9060
CWXDA0015I: Federating the agent, check logfile
   C:\WebSphere\XDAgent\logs\federateAgent.log for federate errors.
CWXDA0018I: Tool actions are being logged in file
C:\WebSphere\XDAgent\logs\federateAgent.log.
CWXDA0014I: Changing agent name to local host name from Default.
CWXDA0006I: Agent hostB is successfully federated into the cell.
```

Note: the addAgent will not use any settings in the hosts file for the name but instead takes the Computer Name as the default node name for the agent. If the computer name is not fully qualified, then it will use the IP address for the node name.

____ 8. Issue this command to start the agent:

startAgent

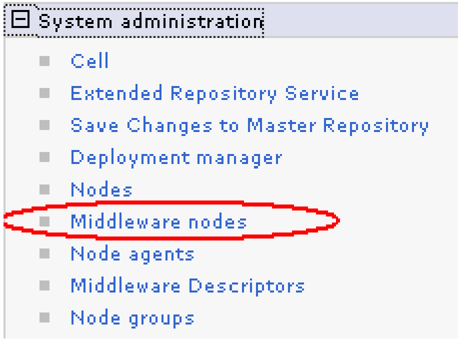
Note: alternatively, you can start the agent from the administrative console by performing the operational action of “Start agent” in the Middleware nodes panel. You must know the “Remote node user ID” and “Remote node user password” for the node where the agent is located.

____ 9. Repeat steps 1 through 3 to create, federate and start the agent for hostC

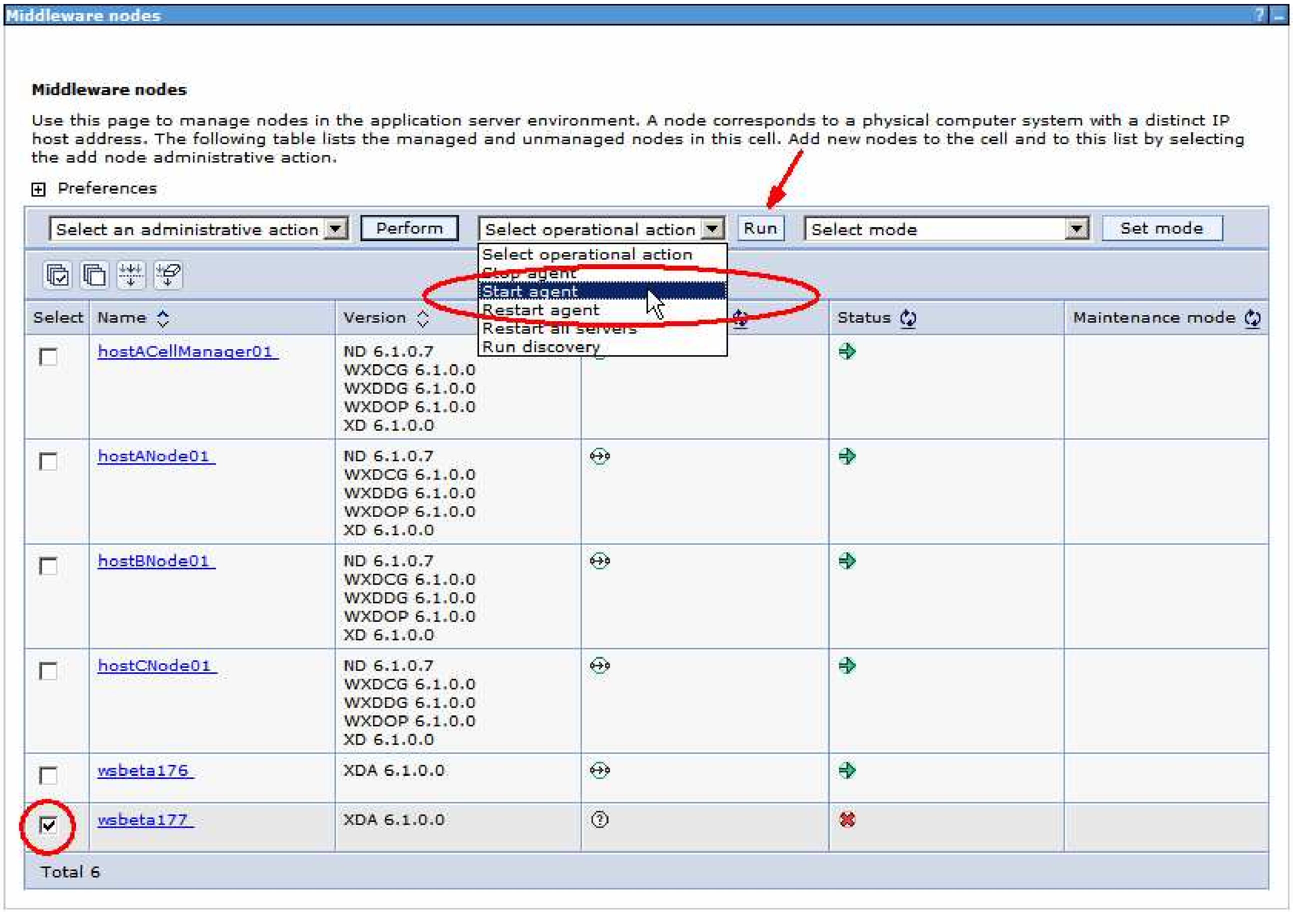
____ 10. Log on to the WebSphere administrative console (if you were logged on before the agents were added, log off and log on again):

http://hostA:9060/ibm/console

____ 11. Review the two new unmanaged nodes within the administrative console. Expand **System administration**, then click on **Middleware Nodes**.



____ 12. The **Middleware Nodes** list should appear. Both agents should be active. If they are not, the select the agent or agents that are inactive, then select **Start Agent** in the dropdown box, and then click **Run**.



Middleware nodes

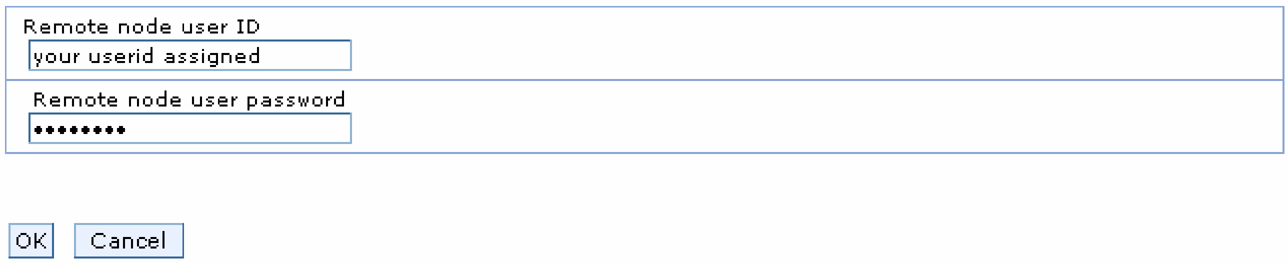
Use this page to manage nodes in the application server environment. A node corresponds to a physical computer system with a distinct IP host address. The following table lists the managed and unmanaged nodes in this cell. Add new nodes to the cell and to this list by selecting the add node administrative action.

⊞ Preferences

Select	Name	Version		Status	Maintenance mode
□	hostACellManager01	ND 6.1.0.7 WXDCG 6.1.0.0 WXDDG 6.1.0.0 WXDOP 6.1.0.0 XD 6.1.0.0		➜	
□	hostANode01	ND 6.1.0.7 WXDCG 6.1.0.0 WXDDG 6.1.0.0 WXDOP 6.1.0.0 XD 6.1.0.0	⊕	➜	
□	hostBNode01	ND 6.1.0.7 WXDCG 6.1.0.0 WXDDG 6.1.0.0 WXDOP 6.1.0.0 XD 6.1.0.0	⊕	➜	
□	hostCNode01	ND 6.1.0.7 WXDCG 6.1.0.0 WXDDG 6.1.0.0 WXDOP 6.1.0.0 XD 6.1.0.0	⊕	➜	
□	wsbeta176	XDA 6.1.0.0	⊕	➜	
☑	wsbeta177	XDA 6.1.0.0	?	✖	

Total 6

Starting an agent requires a valid user ID and password for node where the agent or node agent is located. Type in a valid user ID and password and click **OK**.



Remote node user ID: your userid assigned

Remote node user password: ••••••••

Note: It is possible that the agent will not be completely started after the “Start agent” request completes. If the agent status still shows stopped, wait 30 seconds and refresh the display.

Non WAS Dynamic Operations

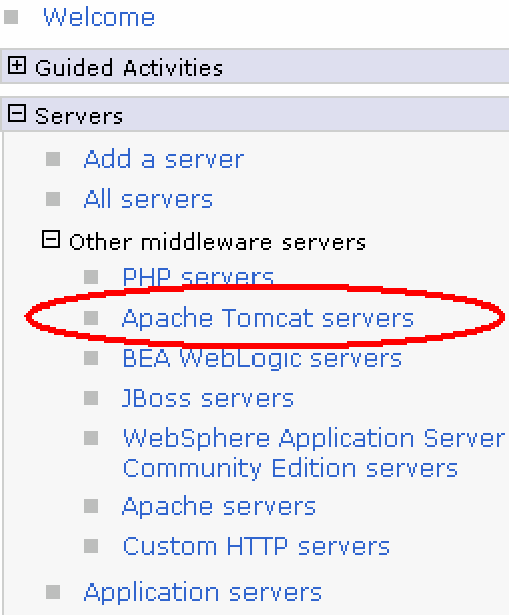
Part 4: Define the Apache Tomcat servers

____ 13. Log on to the WebSphere administrative console

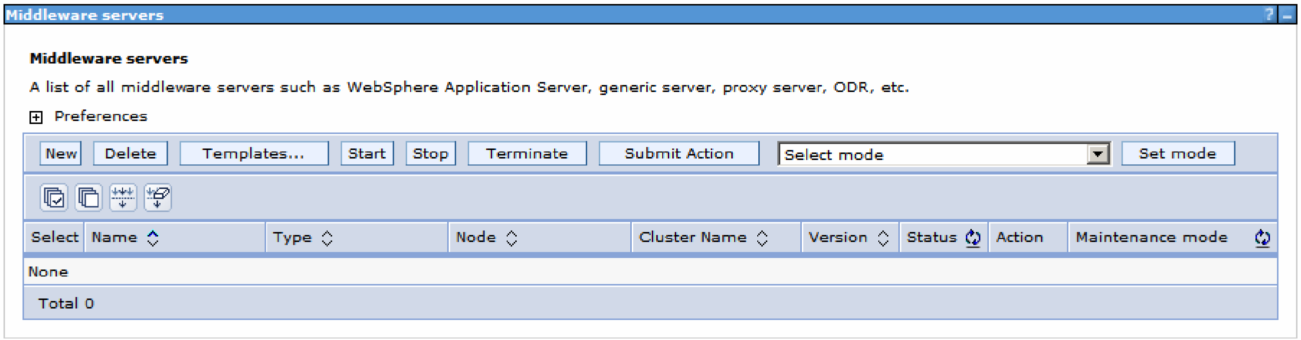
http://hostA:9060/ibm/console

____ 14. Define the Apache Tomcat server on hostB to WebSphere.

__ a. In the left panel of the administrative console, expand **Servers**, then expand **Other middleware servers.**



__ b. Click on **Apache Tomcat servers.**



Middleware servers

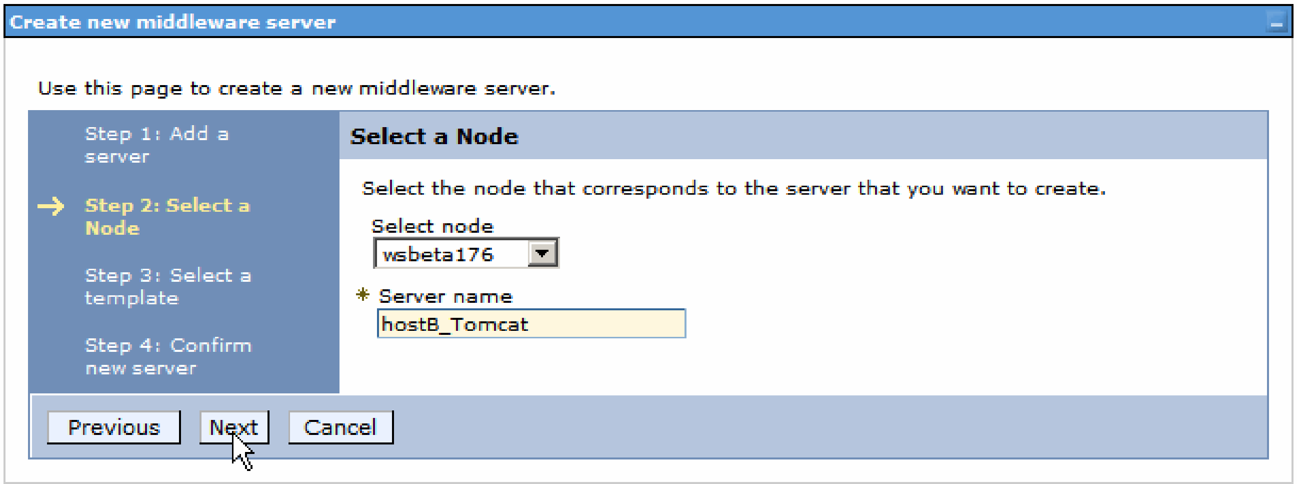
A list of all middleware servers such as WebSphere Application Server, generic server, proxy server, ODR, etc.

⊞ Preferences

Select	Name	Type	Node	Cluster Name	Version	Status	Action	Maintenance mode
None								
Total 0								

__ c. On the Middleware servers panel, click **New.**

__ d. On the **Step 2: Select a Node panel**, select **hostB** or equivalent node in the node dropdown list and type in the Server name **hostB_Tomcat**. Take care that you select a WebSphere agent rather than a node agent, since both will be seen in the node dropdown list. Tomcat server definitions can be associated with WebSphere node agents, but this lab demonstrates the use of Tomcat servers with WebSphere agents, which have a much smaller footprint than WebSphere node agents.


Create new middleware server
Step 1: Add a server
Step 2: Select a Node
Step 3: Select a template
Step 4: Confirm new server
Previous
Next
Cancel

Use this page to create a new middleware server.

Select a Node

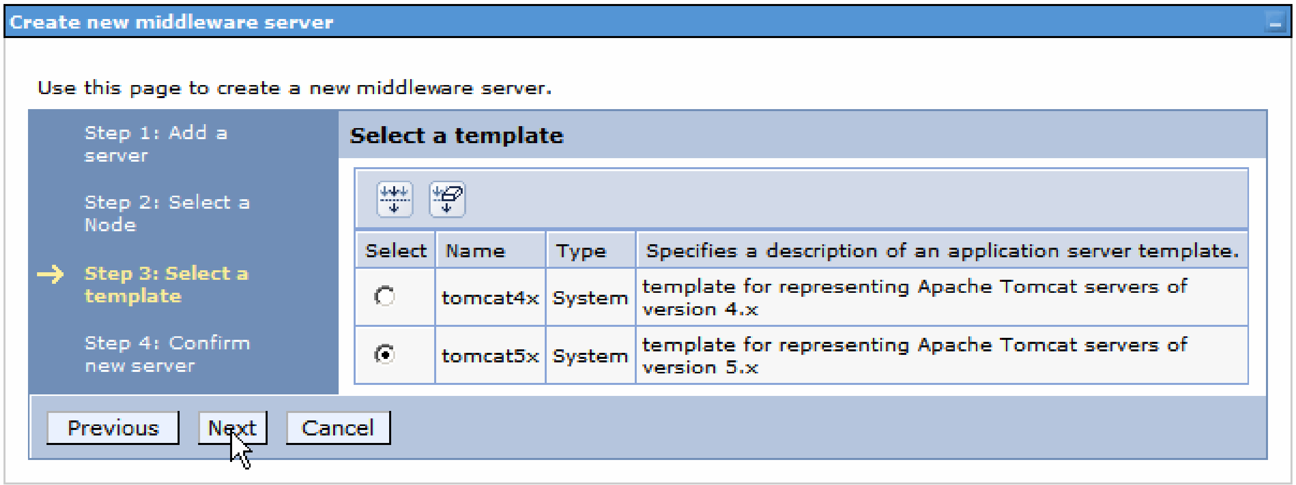
Select the node that corresponds to the server that you want to create.

Select node: wsbeta176

* Server name: hostB_Tomcat

__ e. Click **Next**.

__ f. Ensure that template **tomcat5x** is selected as template for the Tomcat server, and click **Next**.

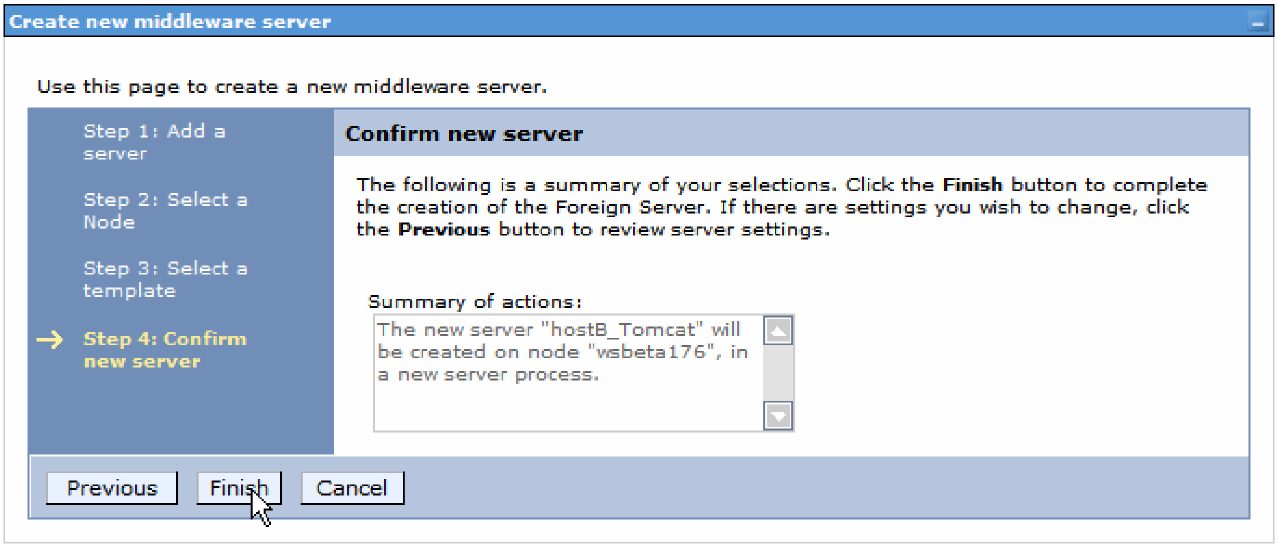

Create new middleware server
Step 1: Add a server
Step 2: Select a Node
Step 3: Select a template
Step 4: Confirm new server
Previous
Next
Cancel

Use this page to create a new middleware server.

Select a template

Select	Name	Type	Specifies a description of an application server template.
○	tomcat4x	System	template for representing Apache Tomcat servers of version 4.x
●	tomcat5x	System	template for representing Apache Tomcat servers of version 5.x

__ g. Review the summary screen, and click **Finish**



Create new middleware server

Use this page to create a new middleware server.

Confirm new server

The following is a summary of your selections. Click the **Finish** button to complete the creation of the Foreign Server. If there are settings you wish to change, click the **Previous** button to review server settings.

Summary of actions:

The new server "hostB_Tomcat" will be created on node "wsbeta176", in a new server process.

____ 15. Save the changes.

__ a. Click **Review** in the messages area.

__ b. On the Save panel, make sure that the check box **Synchronize changes with Nodes** is selected.

__ c. Click **Save.**

__ d. Click **OK** when the sync operation completes

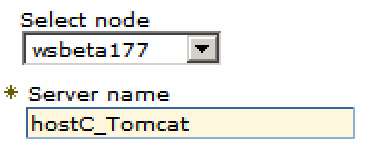
____ 16. Define the Apache Tomcat server on **hostC** to WebSphere.

__ a. In the left panel of the administrative console, expand **Servers**, then expand **Other middleware servers.**

__ b. Click on **Apache Tomcat servers.**

__ c. On the Middleware servers panel, click **New**.

__ d. On the Select a Node panel, select **hostC** or the equivalent node in the node dropdown list and type in the Server name **hostC_Tomcat**.



Select node

wsbeta177

* Server name

hostC_Tomcat

__ e. Click **Next**.

__ f. Ensure that template **tomcat5x** is selected as template for the Tomcat server, and click **Next**.

__ g. Review the summary screen, and click **Finish**.

____ 17. Save the changes.

__ a. Click **Review** in the messages area.

__ b. On the Save panel, make sure that the check box **Synchronize changes with Nodes** is selected.

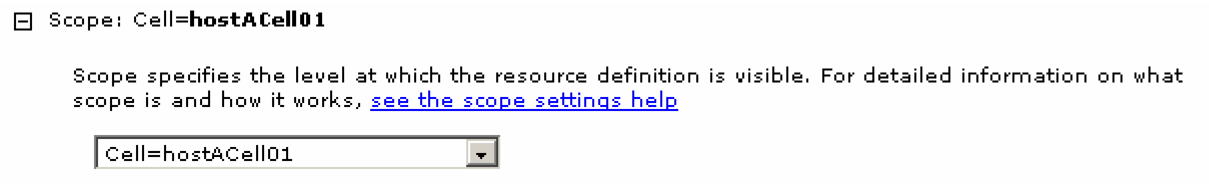
__ c. Click **Save.**

__ d. Click **OK** when the sync operation completes

____ 18. Set the Environmental settings

__ a. In the left panel of the administrative console, click on “**Environment**”, then click on “**WebSphere Variables**”.

__ b. Set the Scope to Cell=hostACell01



⊟ Scope: Cell=**hostACell01**

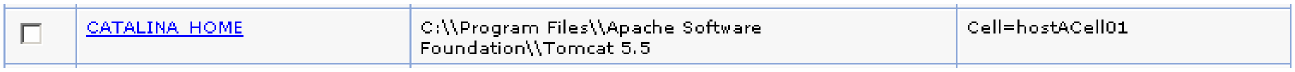
Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help

Cell=hostACell01

__ c. Click on variable **CATALINA_HOME .**

__ d. Set the value of this variable to **C:\\Program Files\\Apache Software Foundation\\Tomcat 5.5.** Ensure the setting matches the actual directory path to your own Apache Tomcat installation. Using the Cell level scope for this environmental variable implies that Tomcat is installed in the same location on both remote nodes. If this is not the case, the set the variable at the node level for each node.

__ e. Click **OK**



□	CATALINA_HOME	C:\\Program Files\\Apache Software Foundation\\Tomcat 5.5	Cell=hostACell01

____ 19. Save the changes.

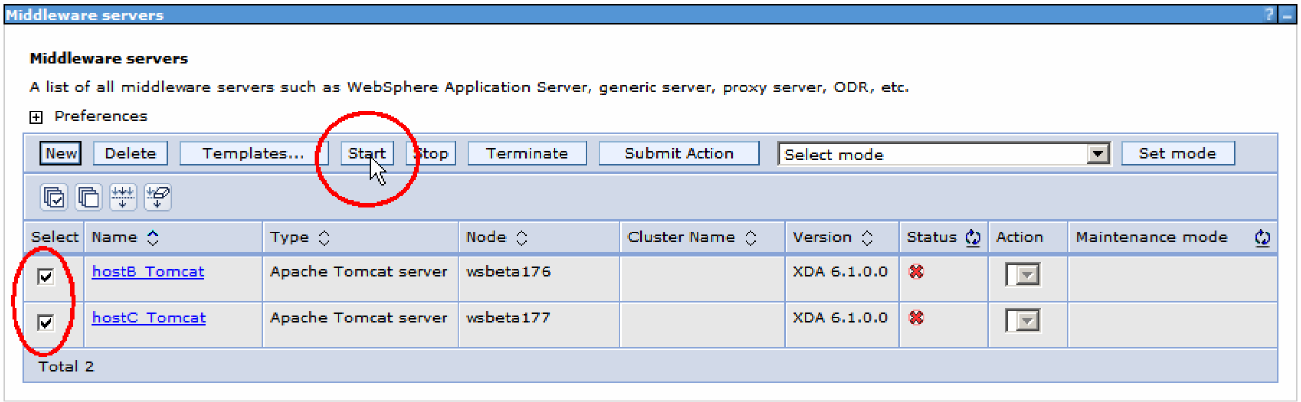
__ a. Click **Review** in the messages area.

__ b. On the Save panel, make sure that the check box **Synchronize changes with Nodes** is selected.

__ c. Click **Save.**

__ d. Click **OK** when the sync operation completes

____ 20. Verify the hostB_Tomcat and hostC_Tomcat servers will start. Select the two servers, then click **Start**. (If the start does not succeed the first time, wait 30 seconds and attempt the start again, to ensure that all environmental changes are replicated to the agent nodes.)



Middleware servers

Middleware servers

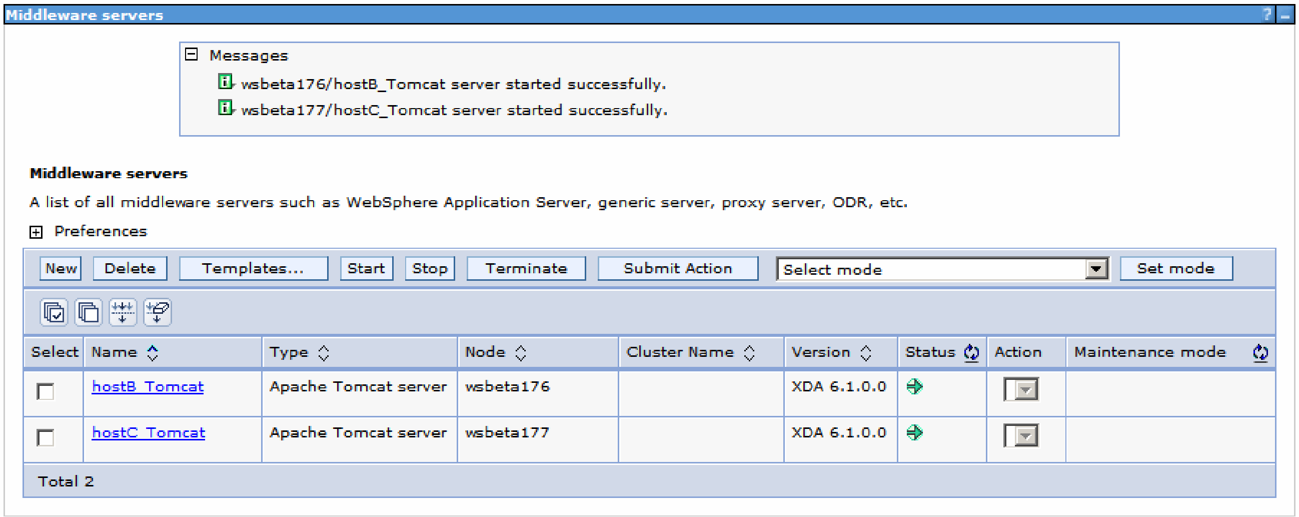
A list of all middleware servers such as WebSphere Application Server, generic server, proxy server, ODR, etc.

⊞ Preferences

Select	Name	Type	Node	Cluster Name	Version	Status	Action	Maintenance mode
☑	hostB_Tomcat	Apache Tomcat server	wsbeta176		XDA 6.1.0.0	✖		
☑	hostC_Tomcat	Apache Tomcat server	wsbeta177		XDA 6.1.0.0	✖		
Total 2								

____ 21. You should see the successful start below.

Note: The start from the administrative console in WebSphere Extended Deployment does NOT start Tomcat as a Windows service. Therefore you should not start Tomcat manually using the Windows Services menu on the node where Tomcat is installed. If Tomcat is running as a Windows service on that node, stop the service and then issue the start from the administrative console again.



Middleware servers

⊟ Messages

wsbeta176/hostB_Tomcat server started successfully.

wsbeta177/hostC_Tomcat server started successfully.

Middleware servers

A list of all middleware servers such as WebSphere Application Server, generic server, proxy server, ODR, etc.

⊞ Preferences

Select	Name	Type	Node	Cluster Name	Version	Status	Action	Maintenance mode
□	hostB_Tomcat	Apache Tomcat server	wsbeta176		XDA 6.1.0.0	➜		
□	hostC_Tomcat	Apache Tomcat server	wsbeta177		XDA 6.1.0.0	➜		
Total 2								

____ 22. Stop the two Tomcat servers.

Part 5: Create and configure dynamic clusters

____ 23. Open the administrative console.

__ a. On **hostA**, open a Web browser.

__ b. Enter the URL: http://localhost:9060/ibm/console.

__ c. Enter a userID of your choice and click **Log In.**

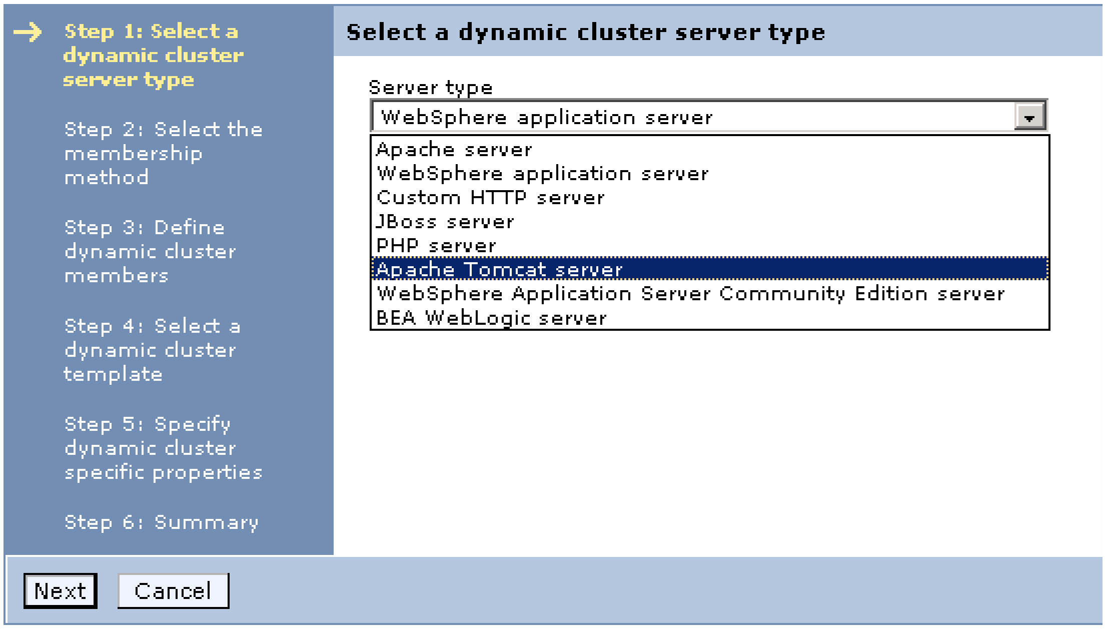
____ 24. Create a dynamic cluster.

__ a. In the administrative console, expand **Servers.**

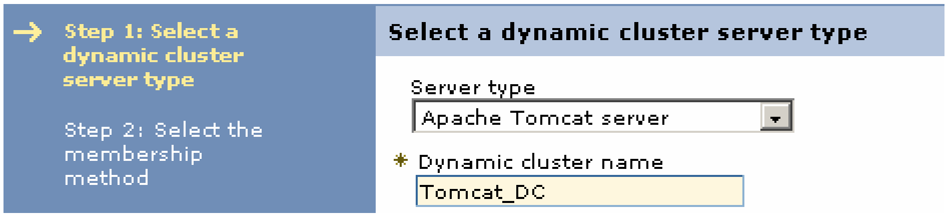
__ b. Click **Dynamic clusters.**

__ c. Click **New.**

__ d. Select the Server type **Apache Tomcat server**.


Step 1: Select a dynamic cluster server type
Step 2: Select the membership method
Step 3: Define dynamic cluster members
Step 4: Select a dynamic cluster template
Step 5: Specify dynamic cluster specific properties
Step 6: Summary
Select a dynamic cluster server type
Server type
WebSphere application server
Apache server
WebSphere application server
Custom HTTP server
JBoss server
PHP server
Apache Tomcat server
WebSphere Application Server Community Edition server
BEA WebLogic server
Next
Cancel

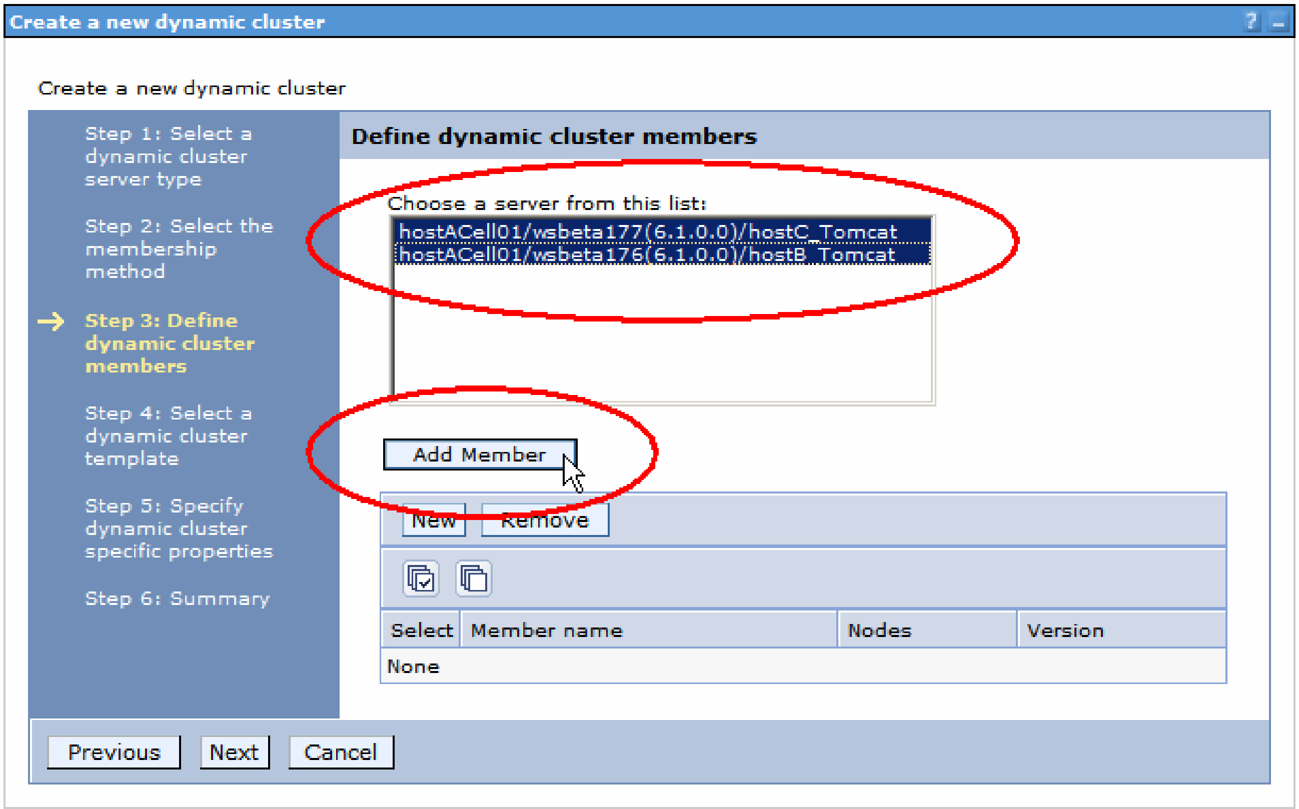
__ e. For Dynamic cluster name, type **Tomcat_DC**, and then click **Next**.


Step 1: Select a dynamic cluster server type
Step 2: Select the membership method
Select a dynamic cluster server type

Server type: Apache Tomcat server

* Dynamic cluster name: Tomcat_DC

__ f. Highlight hostB and hostC servers in the list and click on the **Add Member** button.



Create a new dynamic cluster

Create a new dynamic cluster

Step 1: Select a dynamic cluster server type

Step 2: Select the membership method

→ **Step 3: Define dynamic cluster members**

Step 4: Select a dynamic cluster template

Step 5: Specify dynamic cluster specific properties

Step 6: Summary

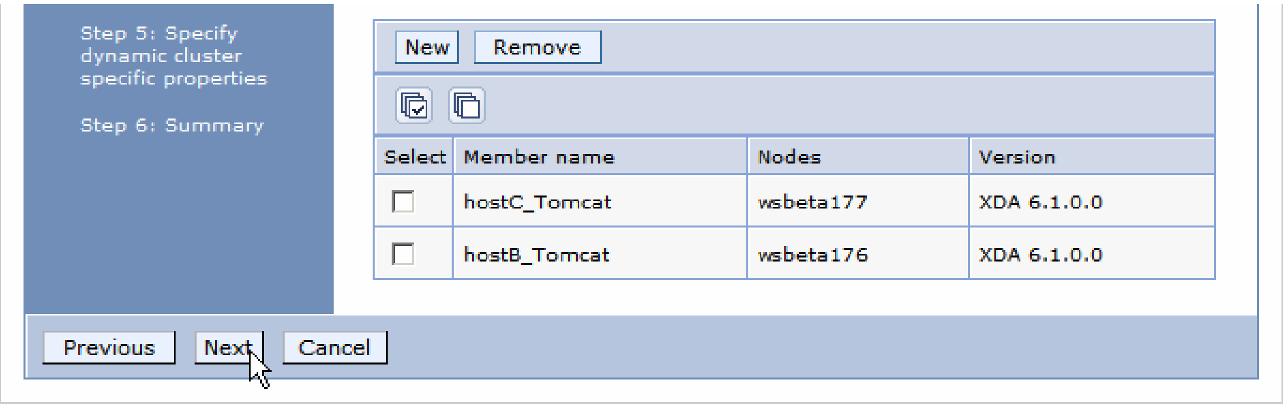
Define dynamic cluster members

Choose a server from this list:

```
hostACell01/wsbeta177(6.1.0.0)/hostC_Tomcat
hostACell01/wsbeta176(6.1.0.0)/hostB_Tomcat
```

Select	Member name	Nodes	Version
None			

__ g. The two members should appear within the list of dynamic cluster members at the bottom.



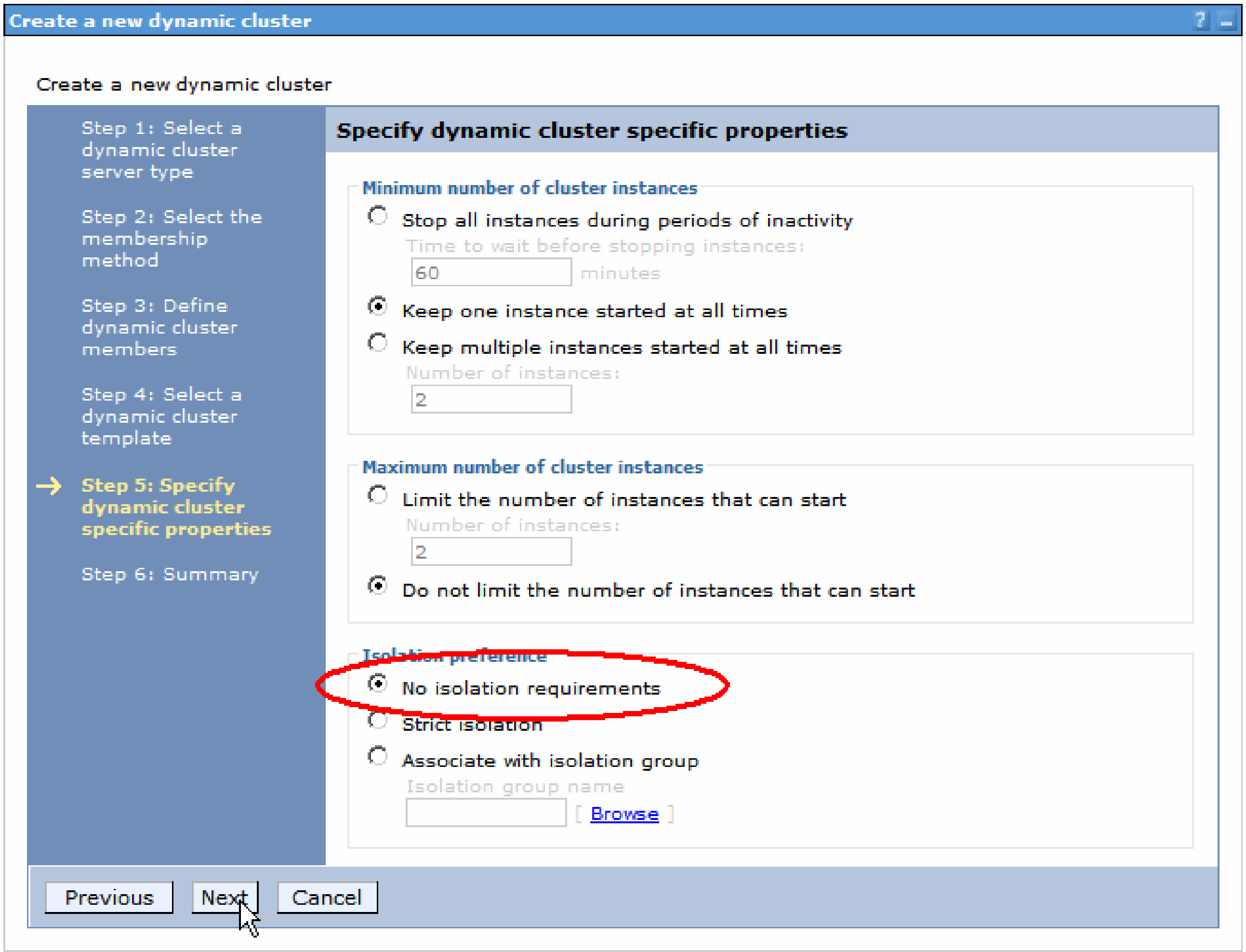
Step 5: Specify dynamic cluster specific properties

Step 6: Summary

Select	Member name	Nodes	Version
□	hostC_Tomcat	wsbeta177	XDA 6.1.0.0
□	hostB_Tomcat	wsbeta176	XDA 6.1.0.0

__ h. Click **Next**.

__ i. In **Step 5 Specify dynamic cluster specific properties,** ensure that the Isolation preference is set to **No isolation requirements.** Accept the remainder of the default settings and click **Next**.



Create a new dynamic cluster

Create a new dynamic cluster

Step 1: Select a dynamic cluster server type

Step 2: Select the membership method

Step 3: Define dynamic cluster members

Step 4: Select a dynamic cluster template

→ Step 5: Specify dynamic cluster specific properties

Step 6: Summary

Specify dynamic cluster specific properties

Minimum number of cluster instances

○ Stop all instances during periods of inactivity

Time to wait before stopping instances: 60 minutes

● Keep one instance started at all times

○ Keep multiple instances started at all times

Number of instances: 2

Maximum number of cluster instances

○ Limit the number of instances that can start

Number of instances: 2

● Do not limit the number of instances that can start

Isolation preference

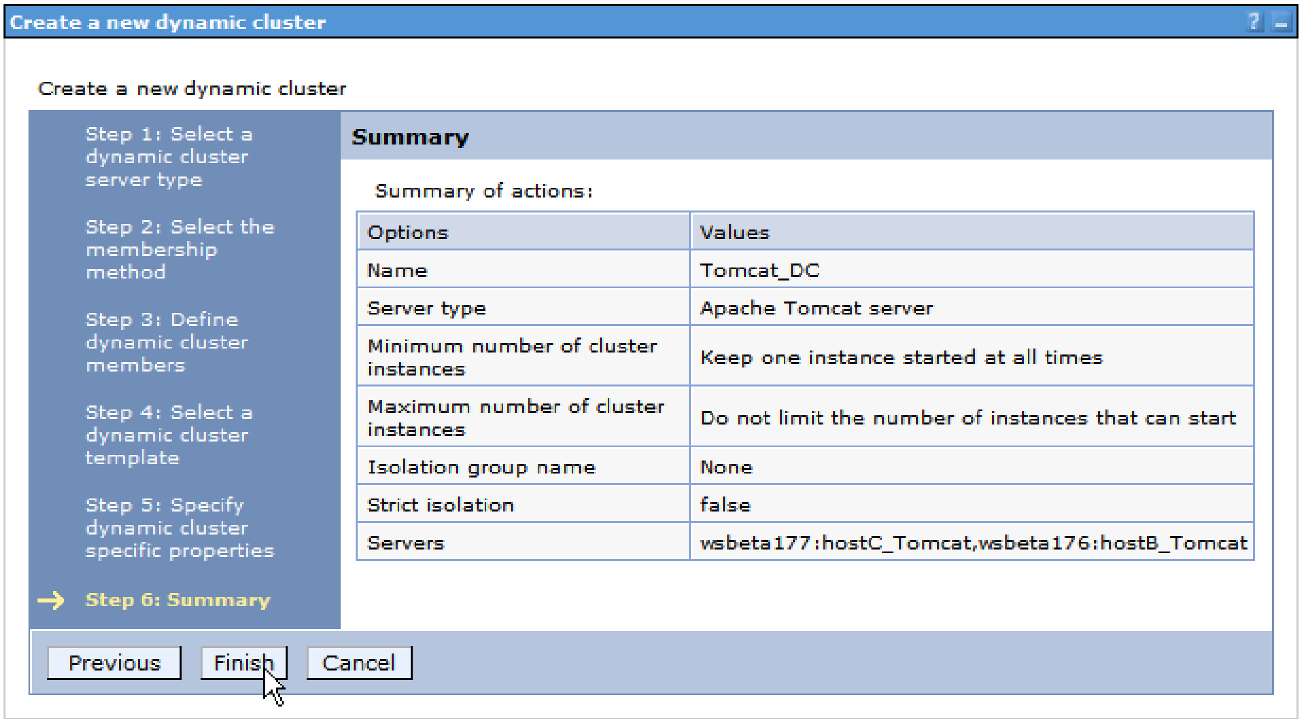
● No isolation requirements

○ Strict isolation

○ Associate with isolation group

Isolation group name [Browse]

__ j. Review the Summary screen, and then click **Finish.**



Create a new dynamic cluster
Step 1: Select a dynamic cluster server type
Step 2: Select the membership method
Step 3: Define dynamic cluster members
Step 4: Select a dynamic cluster template
Step 5: Specify dynamic cluster specific properties
Step 6: Summary
Previous
Finish
Cancel

Create a new dynamic cluster

Summary

Summary of actions:

Options	Values
Name	Tomcat_DC
Server type	Apache Tomcat server
Minimum number of cluster instances	Keep one instance started at all times
Maximum number of cluster instances	Do not limit the number of instances that can start
Isolation group name	None
Strict isolation	false
Servers	wsbeta177:hostC_Tomcat,wsbeta176:hostB_Tomcat

____ 25. Save the changes

__ a. Click **Review** in the messages area (or under the System Administration menu).

__ b. On the Save panel, select the check box **Synchronize changes with Nodes.**

__ c. Click **Save.**

__ d. Click **OK** when the synchronization operation completes.

Non WAS Dynamic Operations

Non WebSphere dynamic operations

Part 6: Install the MicroWebApp application

____ 26. Open the administrative console.

__ a. On **hostA**, open a Web browser.

__ b. Enter the URL: **http://localhost:9060/ibm/console**.

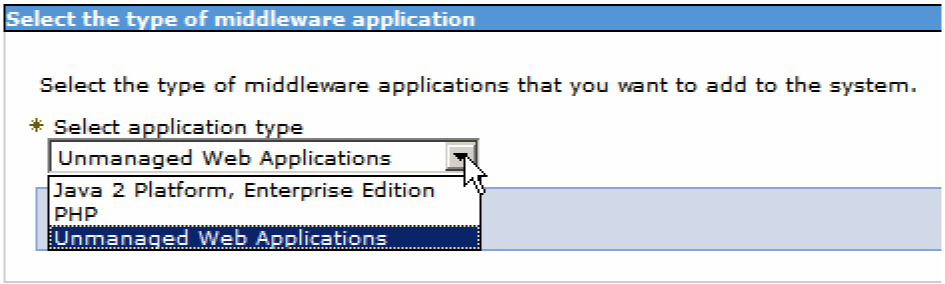
__ c. Enter a userID of your choice and click **Log In.**

____ 27. Install the representation of the MicroWebApp application.

Note: Since WebSphere does not control or affect the actual life cycle or installation of the application on the Tomcat server, WebSphere needs only to know information about the Web modules of the application, so only a representation of the application is required in WebSphere.

__ a. In the Navigation panel, expand **Applications** and click **Install New Middleware Application**.

__ b. In the "**Select Application Type**" download, select **Unmanaged Web Applications,** and then click **Next.**



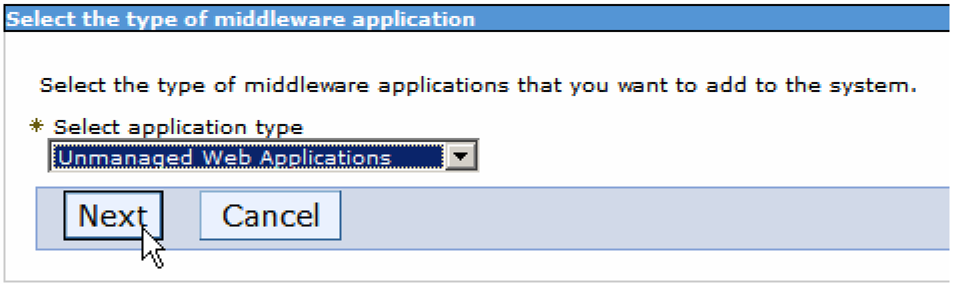
Select the type of middleware application

Select the type of middleware applications that you want to add to the system.

* Select application type

Unmanaged Web Applications

Java 2 Platform, Enterprise Edition
PHP
Unmanaged Web Applications



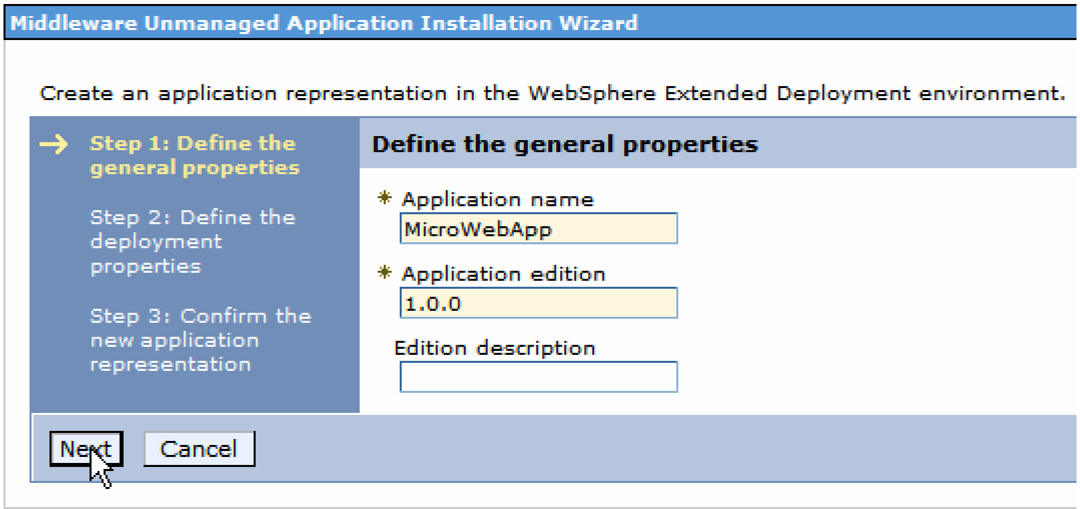
Select the type of middleware application

Select the type of middleware applications that you want to add to the system.

* Select application type

Unmanaged Web Applications

__ c. In the **Step 1: Define the General Properties** panel, type in the application name of **MicroWebApp**. You can leave the default specification for Application Edition, or specify your own. Then click **Next.**



Middleware Unmanaged Application Installation Wizard

Create an application representation in the WebSphere Extended Deployment environment.

→ Step 1: Define the general properties

Step 2: Define the deployment properties

Step 3: Confirm the new application representation

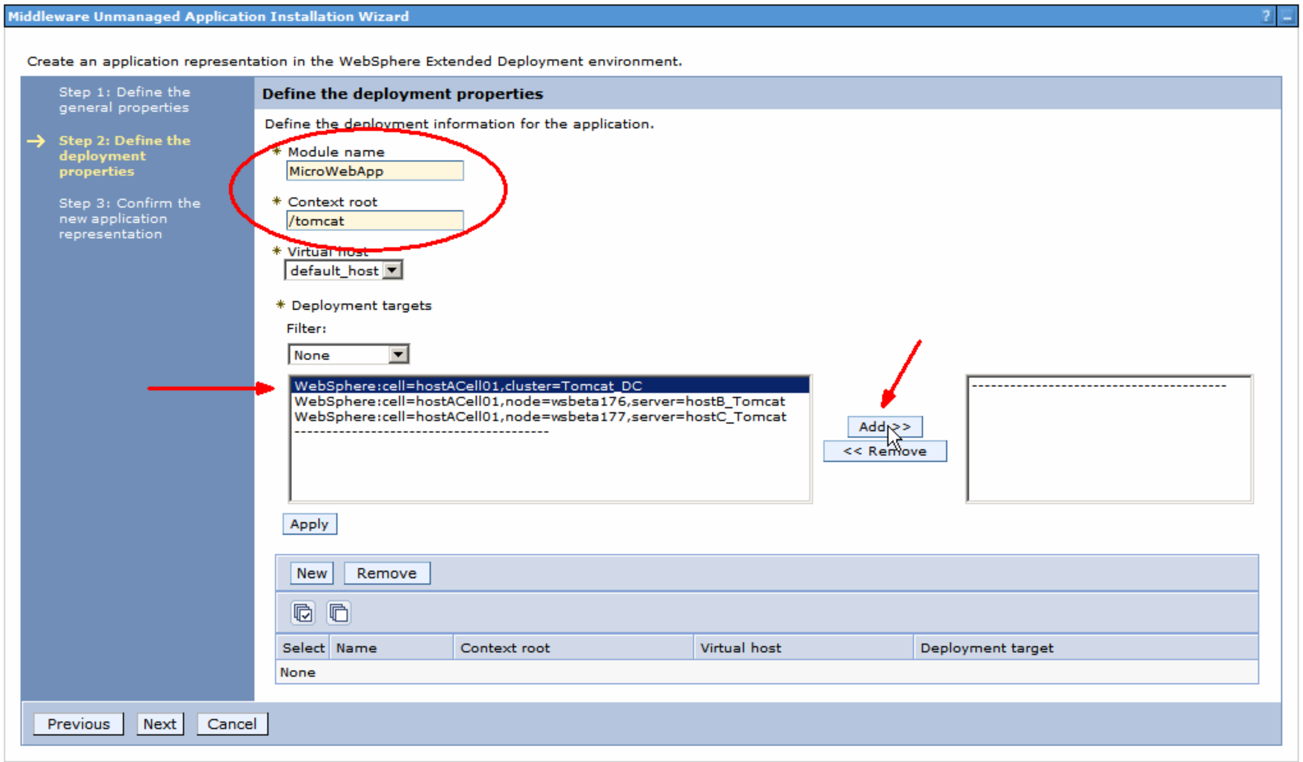
Define the general properties

* Application name: MicroWebApp

* Application edition: 1.0.0

Edition description:

__ d. In the **Step 2: Define the Deployment Properties** panel, type in the Module Name of **MicroWebApp, and** the Context Root of **/tomcat**. Virtual host should be allowed to default to **default_host.** Under Deployment Targets, highlight the **Tomcat_DC** cluster by clicking on it once…



Middleware Unmanaged Application Installation Wizard

Create an application representation in the WebSphere Extended Deployment environment.

Step 1: Define the general properties

→ Step 2: Define the deployment properties

Step 3: Confirm the new application representation

Define the deployment properties

Define the deployment information for the application.

* Module name: MicroWebApp

* Context root: /tomcat

* Virtual host: default_host

* Deployment targets

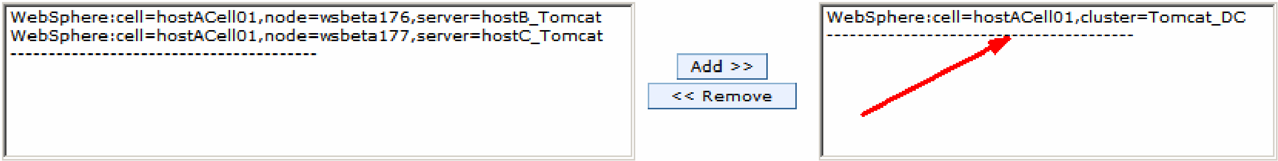
Filter: None

```
WebSphere:cell=hostACell01,cluster=Tomcat_DC
WebSphere:cell=hostACell01,node=wsbeta176,server=hostB_Tomcat
WebSphere:cell=hostACell01,node=wsbeta177,server=hostC_Tomcat
----------------------------------------
```

```
----------------------------------------
```

Select	Name	Context root	Virtual host	Deployment target
None				

__ e. … and then click "**Add**". The cluster should appear in the right pane as a selected target.

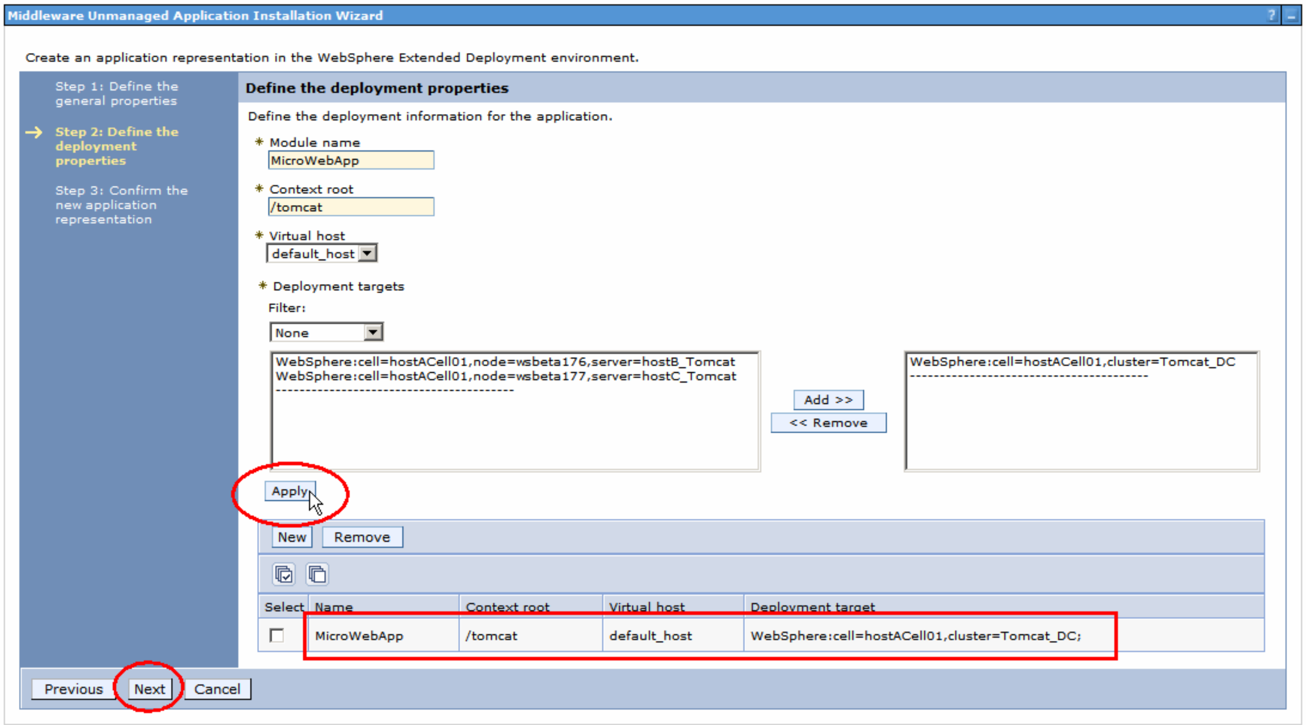


```
WebSphere:cell=hostACell01,node=wsbeta176,server=hostB_Tomcat
WebSphere:cell=hostACell01,node=wsbeta177,server=hostC_Tomcat
----------------------------------------
```

```
WebSphere:cell=hostACell01,cluster=Tomcat_DC
----------------------------------------
```

__ f. Click **Apply**.

The Application should appear within the selection list at the bottom of the panel. (If you had a second application to add, you would click **New** near the bottom and then Add / Apply the second application. In this lab you are adding only a single application.)



Middleware Unmanaged Application Installation Wizard

Create an application representation in the WebSphere Extended Deployment environment.

Step 1: Define the general properties

→ Step 2: Define the deployment properties

Step 3: Confirm the new application representation

Define the deployment properties

Define the deployment information for the application.

* Module name: MicroWebApp

* Context root: /tomcat

* Virtual host: default_host

* Deployment targets

Filter: None

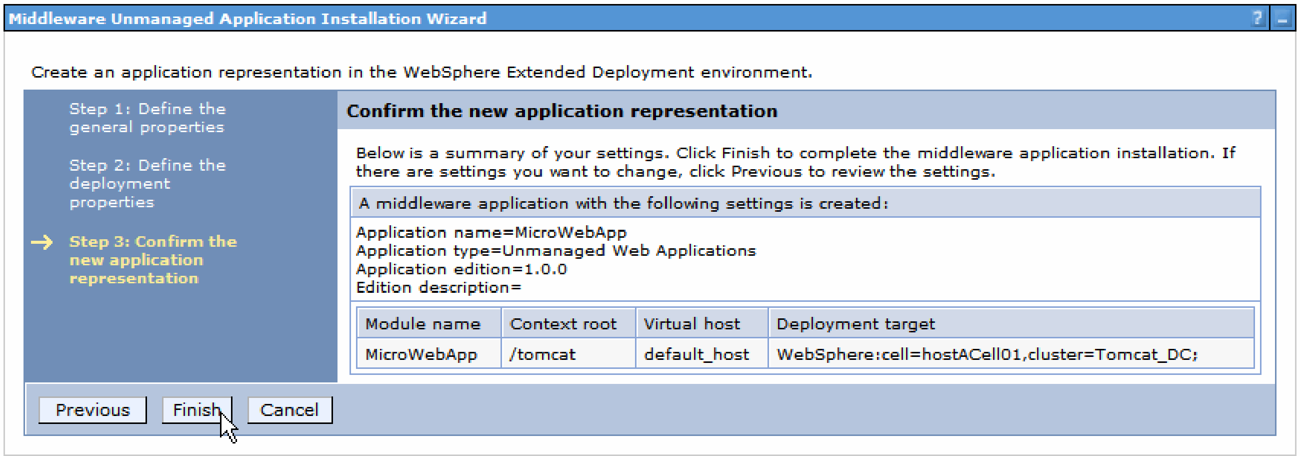
```
WebSphere:cell=hostACell01,node=wsbeta176,server=hostB_Tomcat
WebSphere:cell=hostACell01,node=wsbeta177,server=hostC_Tomcat
----------------------------------------
```

```
WebSphere:cell=hostACell01,cluster=Tomcat_DC
----------------------------------------
```

Select	Name	Context root	Virtual host	Deployment target
□	MicroWebApp	/tomcat	default_host	WebSphere:cell=hostACell01,cluster=Tomcat_DC;

__ g. Click **Next.**

__ h. On **Step 3 Summary**, click **Finish.** Review the information then proceed to the next step.



Middleware Unmanaged Application Installation Wizard

Create an application representation in the WebSphere Extended Deployment environment.

Step 1: Define the general properties

Step 2: Define the deployment properties

→ Step 3: Confirm the new application representation

Confirm the new application representation

Below is a summary of your settings. Click Finish to complete the middleware application installation. If there are settings you want to change, click Previous to review the settings.

A middleware application with the following settings is created:

Application name=MicroWebApp
Application type=Unmanaged Web Applications
Application edition=1.0.0
Edition description=

Module name	Context root	Virtual host	Deployment target
MicroWebApp	/tomcat	default_host	WebSphere:cell=hostACell01,cluster=Tomcat_DC;

____ 28. Save the changes

__ a. Click **Review** in the messages area (or under the System Administration menu).

__ b. On the Save panel, select the check box **Synchronize changes with Nodes.**

__ c. Click **Save.**

__ d. Click **OK** when the synchronization operation completes..

Part 7: Create operational policies

Next, you will create an operational policy, which drives the decisions of the on-demand router (proxy server). An operational policy is made up of a Service Policy and Transaction classes. This policy determines how requests are managed by WebSphere. For this lab exercise, you will create a service policy and transaction class. The mapping between the dynamic cluster, Web module, transaction class, and service policy is depicted in table below.

Dynamic Cluster	Web Module	Transaction Class	Service Policy
Tomcat_DC	MicroWebApp	Tomcat_TC	Tomcat_SP

You will create the service policy Tomcat_SP. The service policy goal is depicted in table below:

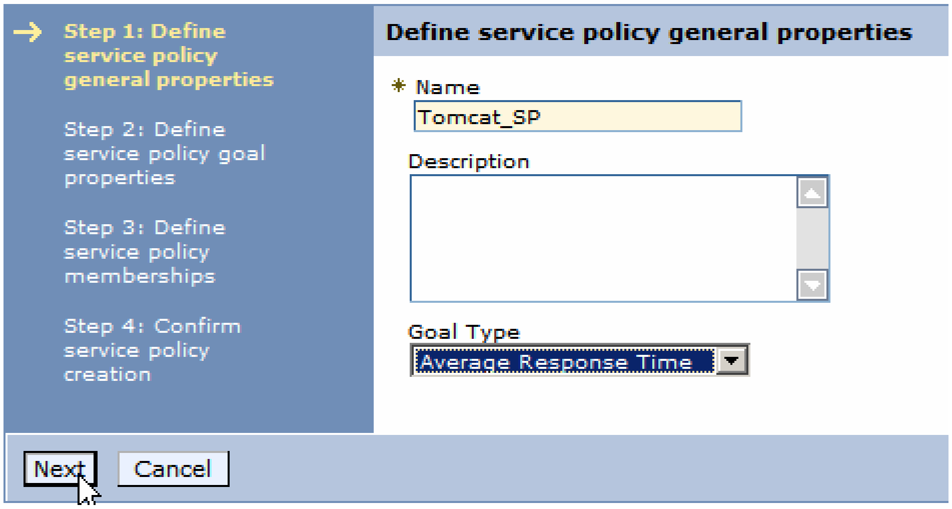
Service Policy	Goal	Importance
Tomcat_SP	500ms	Very high

____ 29. Create the Service policy Tomcat_SP.

__ a. In the Navigation panel, expand **Operational policies** and click **Service policies**.

__ b. Click **New** to create a new service policy.

__ c. Enter the name as **Tomcat_SP**. Select **Average Response Time** from the Goal Type box.



Step 1: Define service policy general properties
Step 2: Define service policy goal properties
Step 3: Define service policy memberships
Step 4: Confirm service policy creation
Next
Cancel

Define service policy general properties

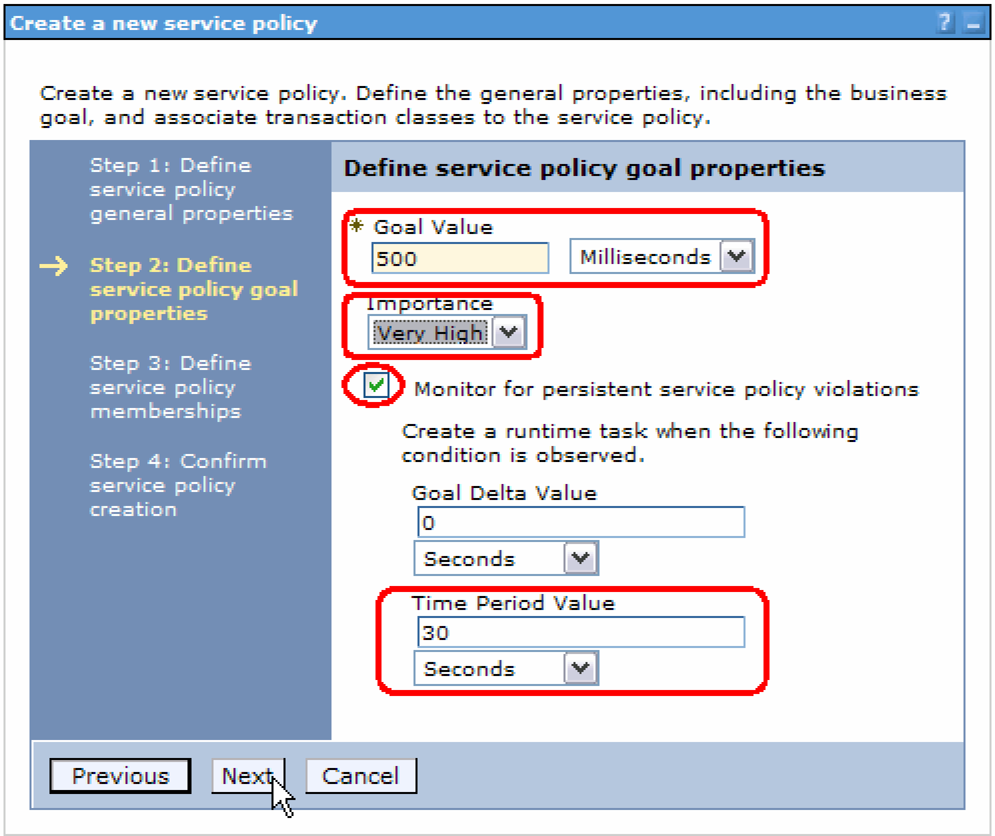
* Name: Tomcat_SP

Description:

Goal Type: Average Response Time

__ d. Click **Next**

__ e. Enter the Goal Value as **500 milliseconds.** Select the Importance as **Very High**. Check the box "Monitor for persistent service policy violations". Set the "Time Period Value" to 30 seconds, so that the goal must be violated for 30 consecutive seconds before monitoring initiates a task. Allow the remainder of the settings to remain as default settings.



Create a new service policy

Create a new service policy. Define the general properties, including the business goal, and associate transaction classes to the service policy.

Step 1: Define service policy general properties

→ **Step 2: Define service policy goal properties**

Step 3: Define service policy memberships

Step 4: Confirm service policy creation

Define service policy goal properties

* Goal Value: 500 Milliseconds

Importance: Very High

☑ Monitor for persistent service policy violations

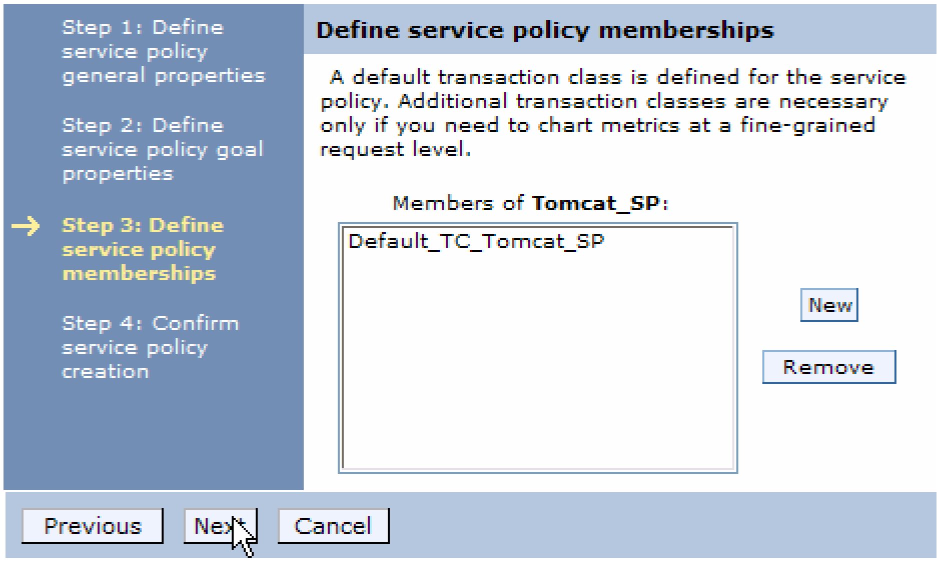
Create a runtime task when the following condition is observed.

Goal Delta Value: 0 Seconds

Time Period Value: 30 Seconds

__ f. Click **Next.**

__ g. In the Memberships panel, note there is a default Transaction Class that appears, and it cannot be removed; you will add a specific Transaction Class later. Click **Next** to accept the default.



Step 1: Define service policy general properties

Step 2: Define service policy goal properties

→ **Step 3: Define service policy memberships**

Step 4: Confirm service policy creation

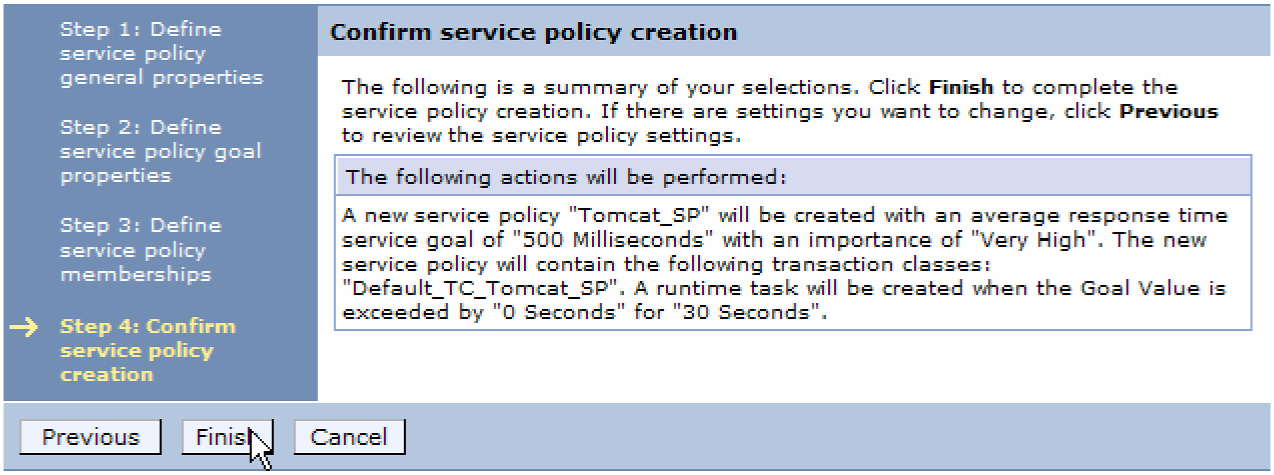
Define service policy memberships

A default transaction class is defined for the service policy. Additional transaction classes are necessary only if you need to chart metrics at a fine-grained request level.

Members of **Tomcat_SP**:

Default_TC_Tomcat_SP

__ h. In the Confirmation panel, click **Finish**.


Step 1: Define service policy general properties
Step 2: Define service policy goal properties
Step 3: Define service policy memberships
Step 4: Confirm service policy creation
Previous
Finish
Cancel

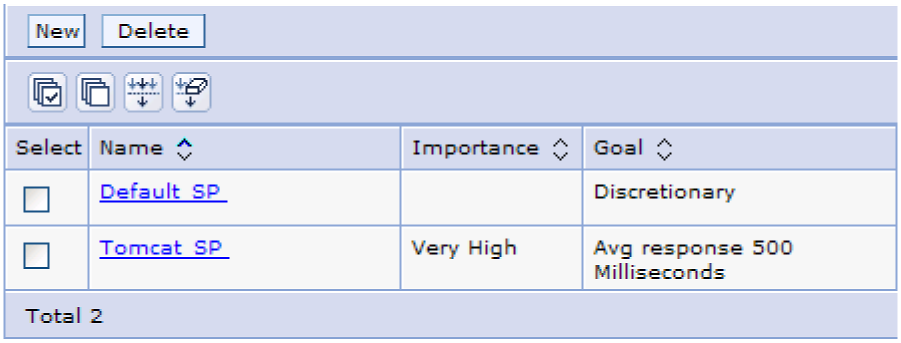
Confirm service policy creation

The following is a summary of your selections. Click **Finish** to complete the service policy creation. If there are settings you want to change, click **Previous** to review the service policy settings.

The following actions will be performed:

A new service policy "Tomcat_SP" will be created with an average response time service goal of "500 Milliseconds" with an importance of "Very High". The new service policy will contain the following transaction classes: "Default_TC_Tomcat_SP". A runtime task will be created when the Goal Value is exceeded by "0 Seconds" for "30 Seconds".

__ i. The **Service Policies** list will look like this.



Select	Name	Importance	Goal
□	Default_SP		Discretionary
□	Tomcat_SP	Very High	Avg response 500 Milliseconds

Total 2

____ 30. Save the changes.

__ a. Click **Review** in the messages area.

__ b. On the Save panel, make sure that the check box **Synchronize changes with Nodes** is selected.

__ c. Click **Save.**

__ d. After the save completes, click **OK**.

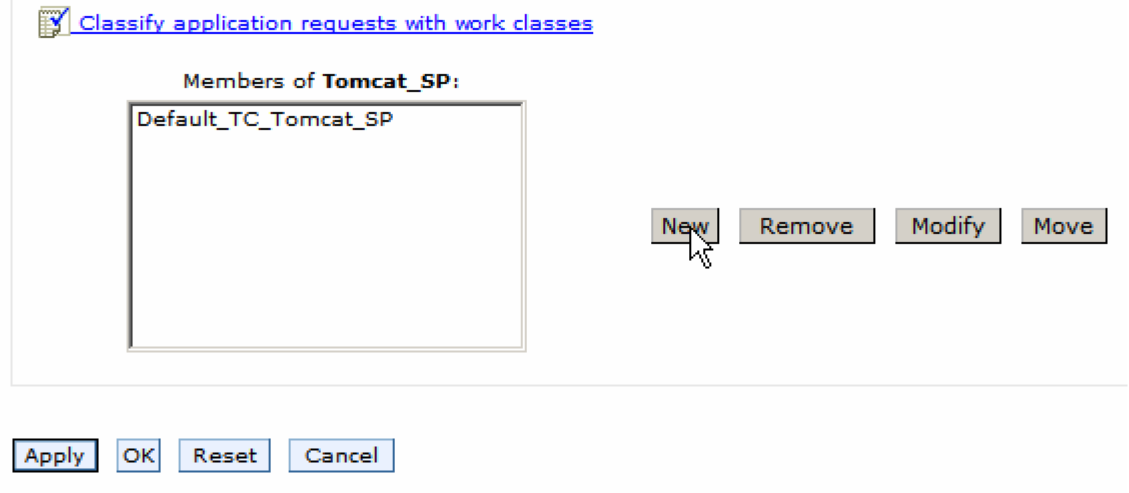
____ 31. Next you will create the Transaction class. You will create the Transaction class Tomcat_TC and map it to the Web module MicroWebApp:

__ a. In the Navigation panel, expand **Operational policies** and click **Service policies**.

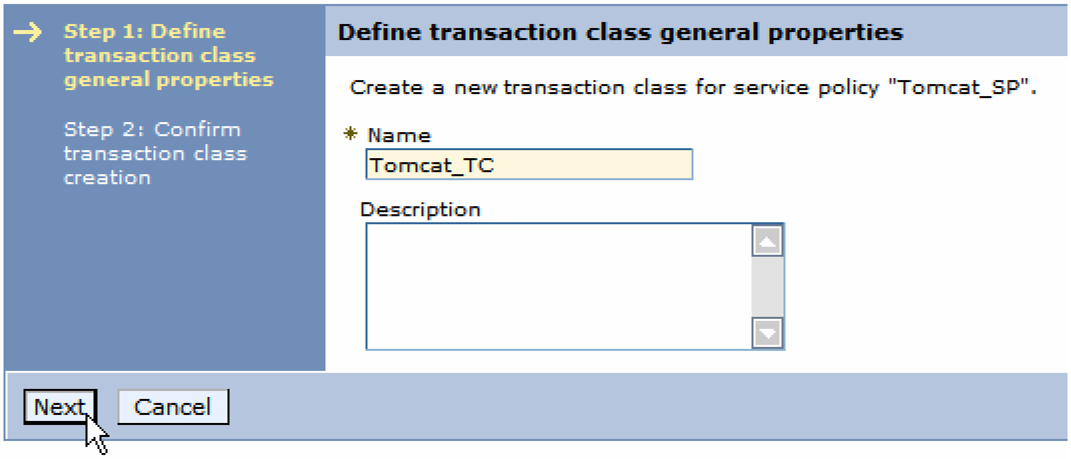
__ b. Click **Tomcat_SP**.

__ c. Near the bottom of the panel, in the Transaction Classes area, click **New** to create a new transaction class. (Note that a default Transaction Class already appears in the window and can be ignored.)

Non WebSphere dynamic operations



__ d. Enter **Tomcat_TC** as the Name.



Step 1: Define transaction class general properties
Step 2: Confirm transaction class creation
Next
Cancel

Define transaction class general properties

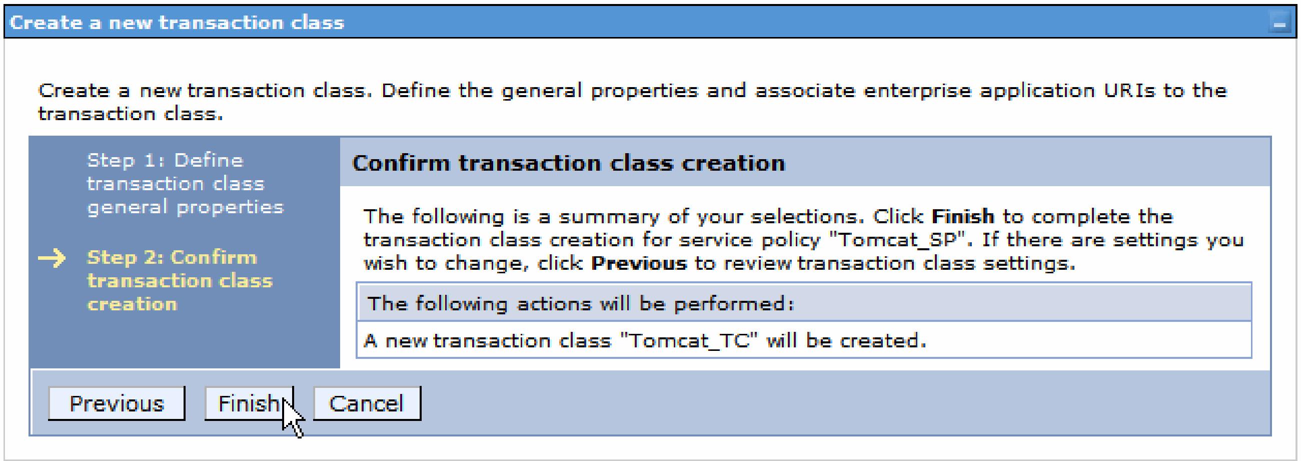
Create a new transaction class for service policy "Tomcat_SP".

* Name: Tomcat_TC

Description

__ e. Click **Next**.

__ f. On the confirmation panel, click **Finish**.



Create a new transaction class
Step 1: Define transaction class general properties
Step 2: Confirm transaction class creation
Previous
Finish
Cancel

Create a new transaction class. Define the general properties and associate enterprise application URIs to the transaction class.

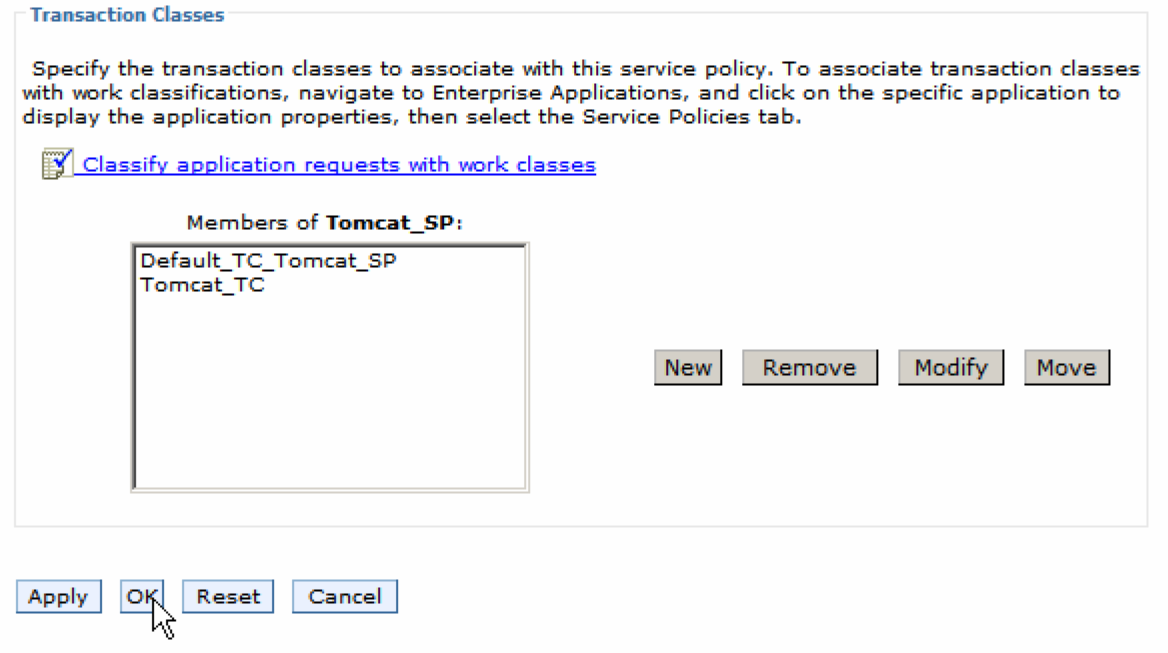
Confirm transaction class creation

The following is a summary of your selections. Click **Finish** to complete the transaction class creation for service policy "Tomcat_SP". If there are settings you wish to change, click **Previous** to review transaction class settings.

The following actions will be performed:

A new transaction class "Tomcat_TC" will be created.

__ g. Click **OK** on the Tomcat_SP Service Policy Configuration page (at the very bottom).



Transaction Classes

Specify the transaction classes to associate with this service policy. To associate transaction classes with work classifications, navigate to Enterprise Applications, and click on the specific application to display the application properties, then select the Service Policies tab.

Classify application requests with work classes

Members of **Tomcat_SP**:

Default_TC_Tomcat_SP
Tomcat_TC

____ 32. Save the changes.

__ a. Click **Review** in the messages area.

__ b. On the Save panel, make sure that the check box **Synchronize changes with Nodes** is selected.

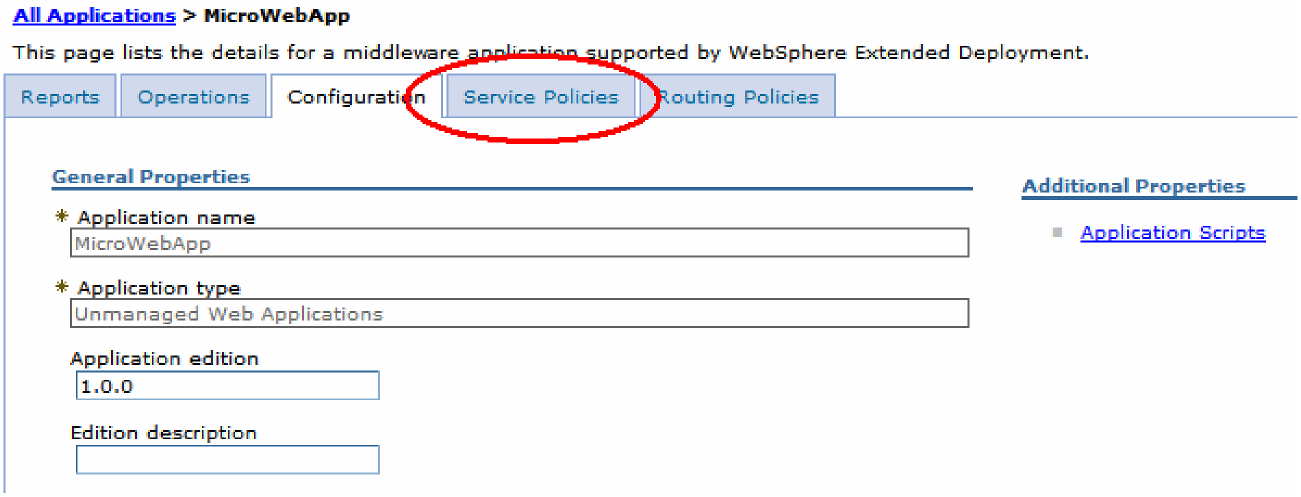
__ c. Click **Save**.

____ 33. Define the **Tomcat_WC** Work Class.

__ a. In the Navigation panel, expand **Applications** and click **All applications.**

__ b. Click on **MicroWebApp.**

__ c. Note the Service Policies tab at the top of the General Properties panel. Click on the **Service Policies** tab.



All Applications > MicroWebApp

This page lists the details for a middleware application supported by WebSphere Extended Deployment.

General Properties

* Application name
MicroWebApp

* Application type
Unmanaged Web Applications

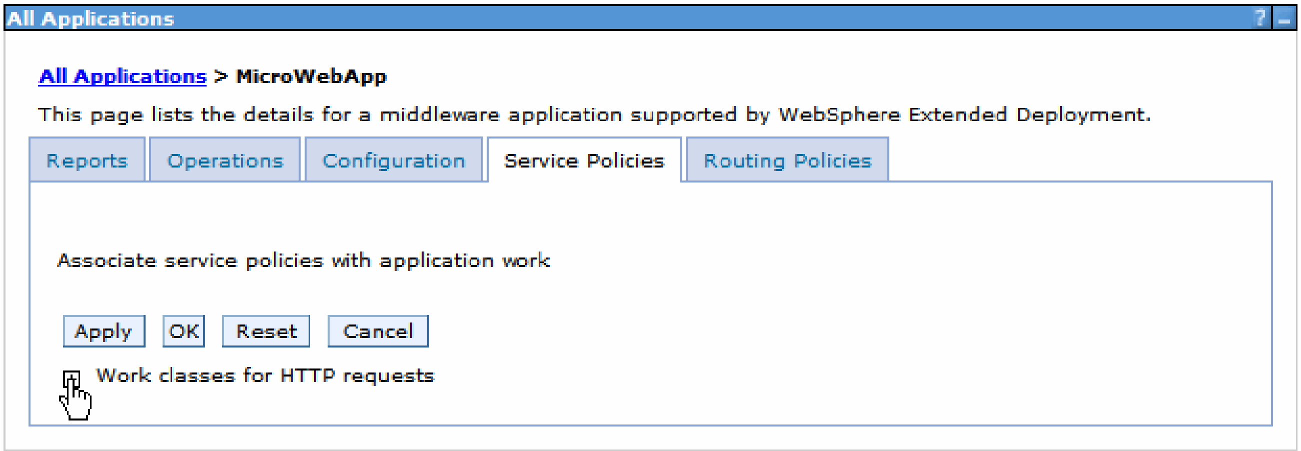
Application edition
1.0.0

Edition description

Additional Properties

- Application Scripts

__ d. Expand **Work Classes For HTTP Requests.**



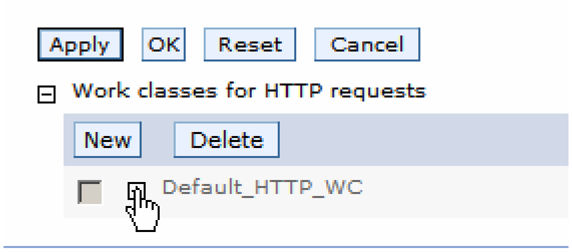
All Applications

All Applications > MicroWebApp

This page lists the details for a middleware application supported by WebSphere Extended Deployment.

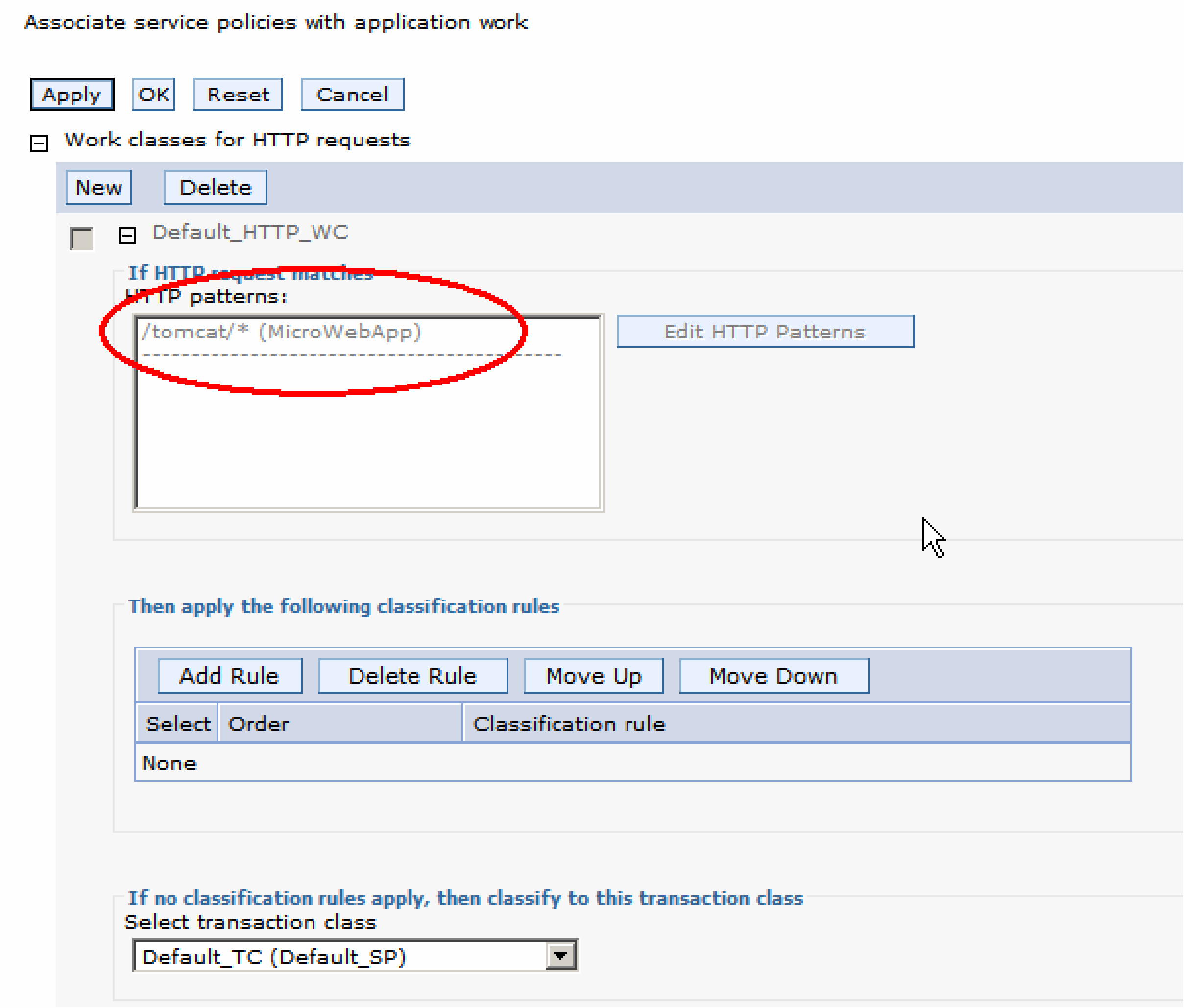
Associate service policies with application work

__ e. Expand **Default_HTTP_WC** (the default work class).



__ f. Note that the HTTP pattern of **/tomcat/*** should already exist by default. If you had specific rules, you could add rules within this panel using the Add Rule classification facilities. But for now, you will associate the /tomcat/* HTTP pattern with your transaction class **Tomcat_TC**.

Non WebSphere dynamic operations



Associate service policies with application work

Work classes for HTTP requests

Default_HTTP_WC

If HTTP request matches

HTTP patterns:

/tomcat/* (MicroWebApp)

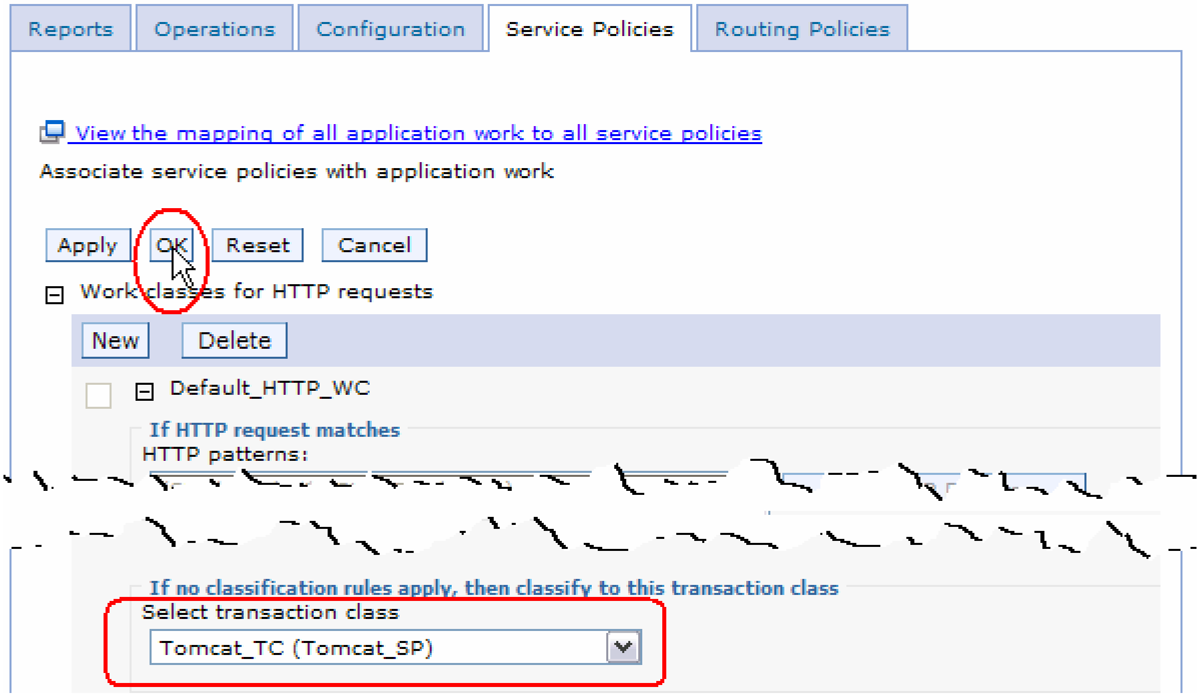
Then apply the following classification rules

Select	Order	Classification rule
None		

If no classification rules apply, then classify to this transaction class

Select transaction class: Default_TC (Default_SP)

__ g. Under **Select transaction class** near the bottom of the panel, select the **Tomcat_TC(Tomcat_SP)** transaction class from the dropdown list. Then click **OK** at the top of the panel.


Reports
Operations
Configuration
Service Policies
Routing Policies
View the mapping of all application work to all service policies
Associate service policies with application work
Apply
OK
Reset
Cancel
Work classes for HTTP requests
New
Delete
Default_HTTP_WC
If HTTP request matches
HTTP patterns:
If no classification rules apply, then classify to this transaction class
Select transaction class
Tomcat_TC (Tomcat_SP)

____ 34. Save the changes.

__ a. Click **Review** in the messages area.

__ b. On the Save panel, make sure that the check box **Synchronize changes with Nodes** is selected.

__ c. Click **Save**.

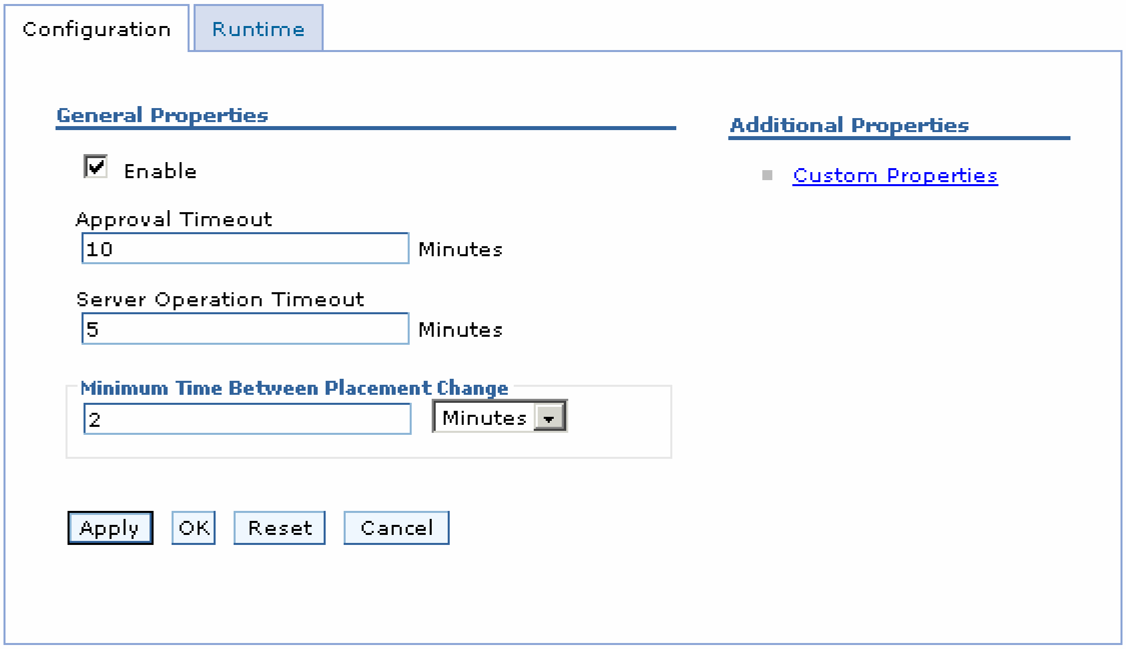
____ 35. The table below contains a mapping of the Work Class names and Transaction Class names, which you have just accomplished.

Work Class Name	Transaction Class Name
Default_HTTP_WC	Tomcat_TC (Tomcat_SP)

____ 36. So that the placement behavior can be observed more quickly than in the default case for the purpose of speeding this exercise, reduce the minimum time between placement changes.

__ a. In the Navigation panel, expand **Operational Policies,** expand **Autonomic Managers,** and click **Application Placement Controller**.

__ b. Set Minimum Time Between Placement Change to **2 Minutes**.



☑ Enable

Approval Timeout: 10 Minutes

Server Operation Timeout: 5 Minutes

Minimum Time Between Placement Change: 2 Minutes

__ c. Click **OK**.

____ 37. Save the changes.

__ a. Click **Review** in the messages area.

__ b. On the Save panel, make sure that the check box **Synchronize changes with Nodes** is selected.

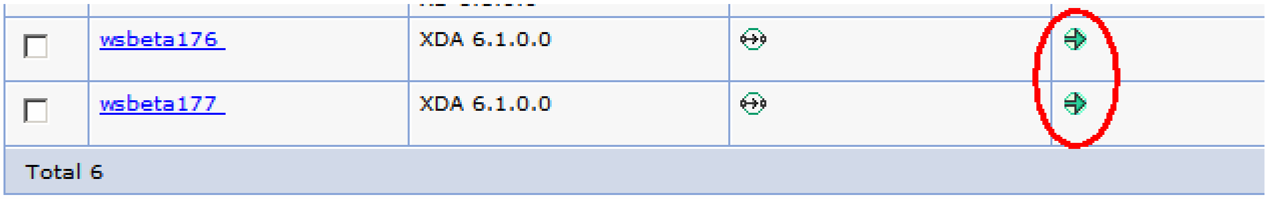
__ c. Click **Save**.

Part 8: Test the application and verify application server placement

____ 38. Check the state of the hostB and hostC middleware nodes.

__ a. In the Navigation panel, expand **System administration**, then click on **Middleware nodes**.

__ b. Ensure that both hostB and hostC WebSphere agents (or the equivalently names agents on your system) are running. You do not need hostBNode01 or hostCNode01 node agents running, since you are not using these federated nodes in your testing in this lab. In the example below, XDAgents wsbeta176 and wsbeta177 are in “Running” state.

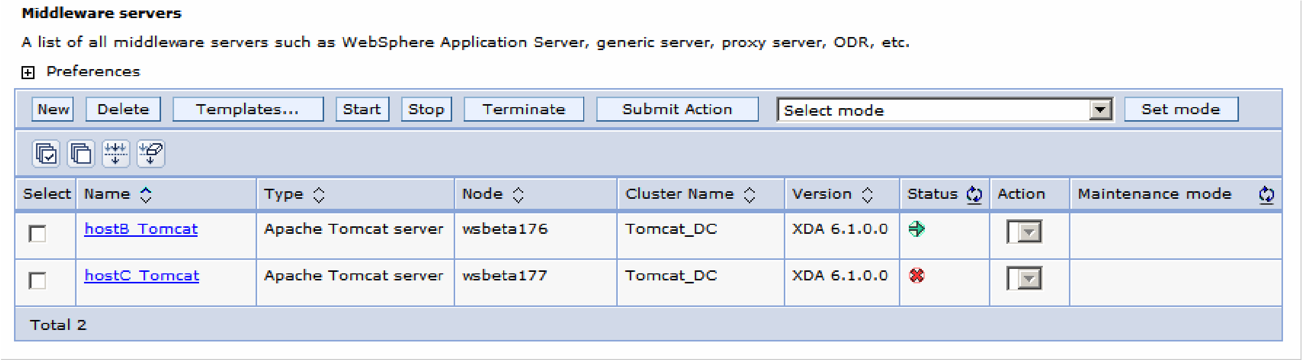


☐	wsbeta176	XDA 6.1.0.0	⊕	➔
☐	wsbeta177	XDA 6.1.0.0	⊕	➔
Total 6				

____ 39. Check the status of Tomcat servers on both the Tomcat nodes and manually set their states..

__ a. In the Navigation panel, expand **Servers**, expand **Middleware Servers,** and click on **Apache Tomcat servers**.

__ b. Note the status of servers instances hostB_Tomcat and hostC_Tomcat servers. The hostB_Tomcat server should be running, and the hostC_Tomcat server should be stopped. If that is not their state, start or stop the appropriate server to set them in this state. This is so you can test placement’s ability to start the additional server when the processor utilization is excessive.



Middleware servers

A list of all middleware servers such as WebSphere Application Server, generic server, proxy server, ODR, etc.

⊞ Preferences

Select	Name	Type	Node	Cluster Name	Version	Status	Action	Maintenance mode
☐	hostB_Tomcat	Apache Tomcat server	wsbeta176	Tomcat_DC	XDA 6.1.0.0	➔		
☐	hostC_Tomcat	Apache Tomcat server	wsbeta177	Tomcat_DC	XDA 6.1.0.0	✖		
Total 2								

____ 40. Start the ODR server if it is not already started.

__ a. In the Navigation panel, expand **Servers** and click **On Demand Routers**.

__ b. Select **ODR** if it is not started, select it and click **Start**.

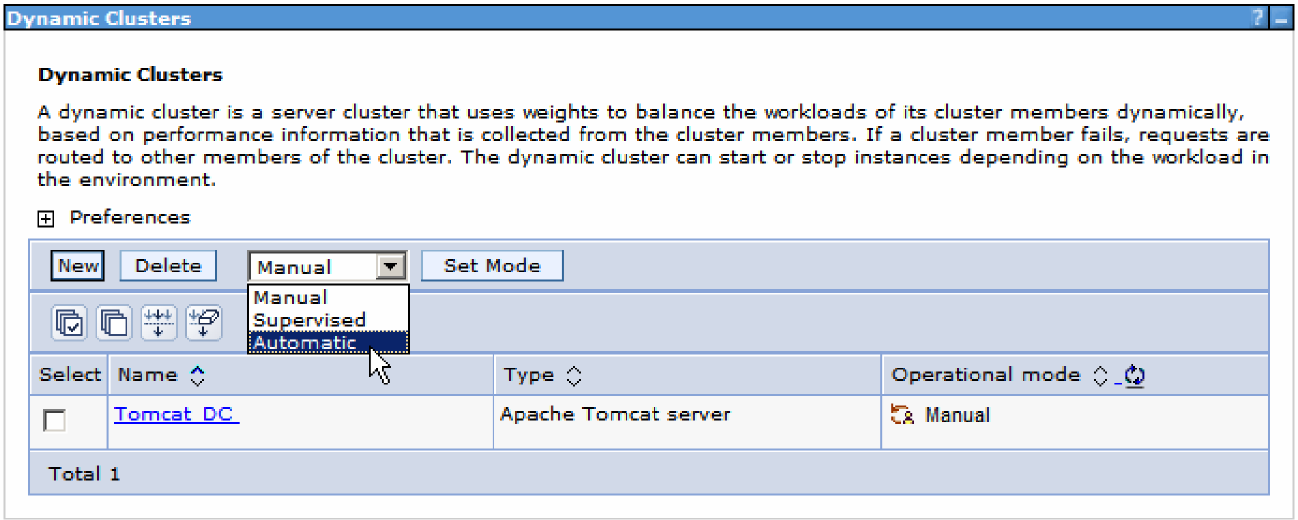
__ c. Wait for confirmation that the server has started.

____ 41. Set the dynamic clusters’ Operational Mode to Automatic.

__ a. In the Navigation panel, expand **Servers** and click **Dynamic Clusters**.

__ b. Select all of the dynamic clusters in the table.

__ c. Select **Automatic** in the drop-down list.

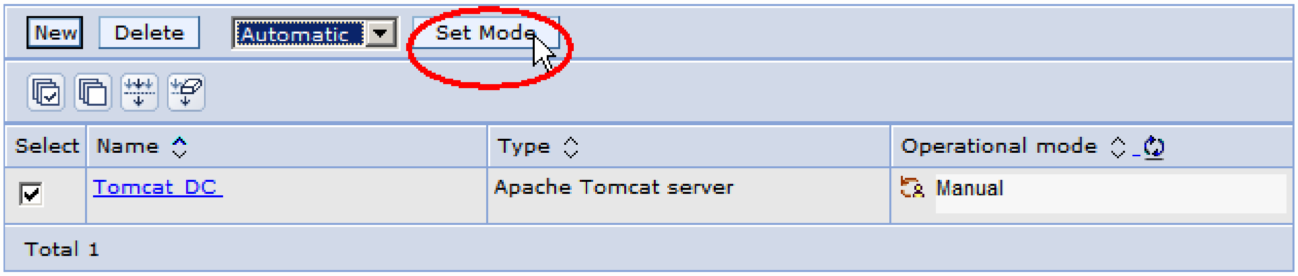


Dynamic Clusters

A dynamic cluster is a server cluster that uses weights to balance the workloads of its cluster members dynamically, based on performance information that is collected from the cluster members. If a cluster member fails, requests are routed to other members of the cluster. The dynamic cluster can start or stop instances depending on the workload in the environment.

Select	Name	Type	Operational mode
□	Tomcat_DC	Apache Tomcat server	Manual
Total 1			

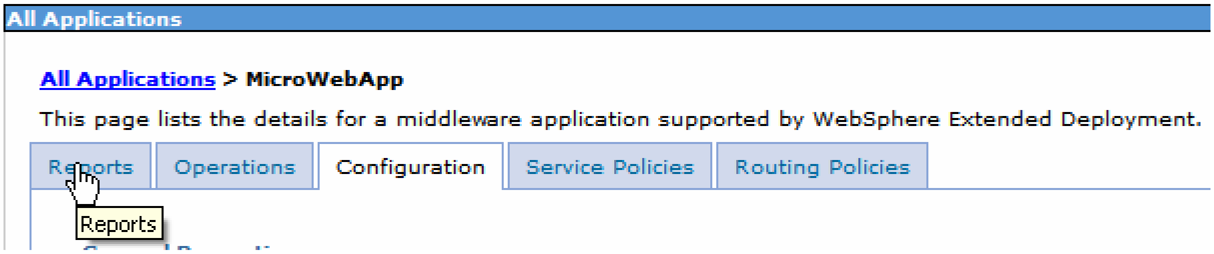
__ d. Click **Set mode**.



Select	Name	Type	Operational mode
☑	Tomcat_DC	Apache Tomcat server	Manual
Total 1			

____ 42. In the Navigation panel, expand **All applications** and click **MicroWebApp**, then click on **Reports.** Accept the Adobe Software License agreement if it appears.

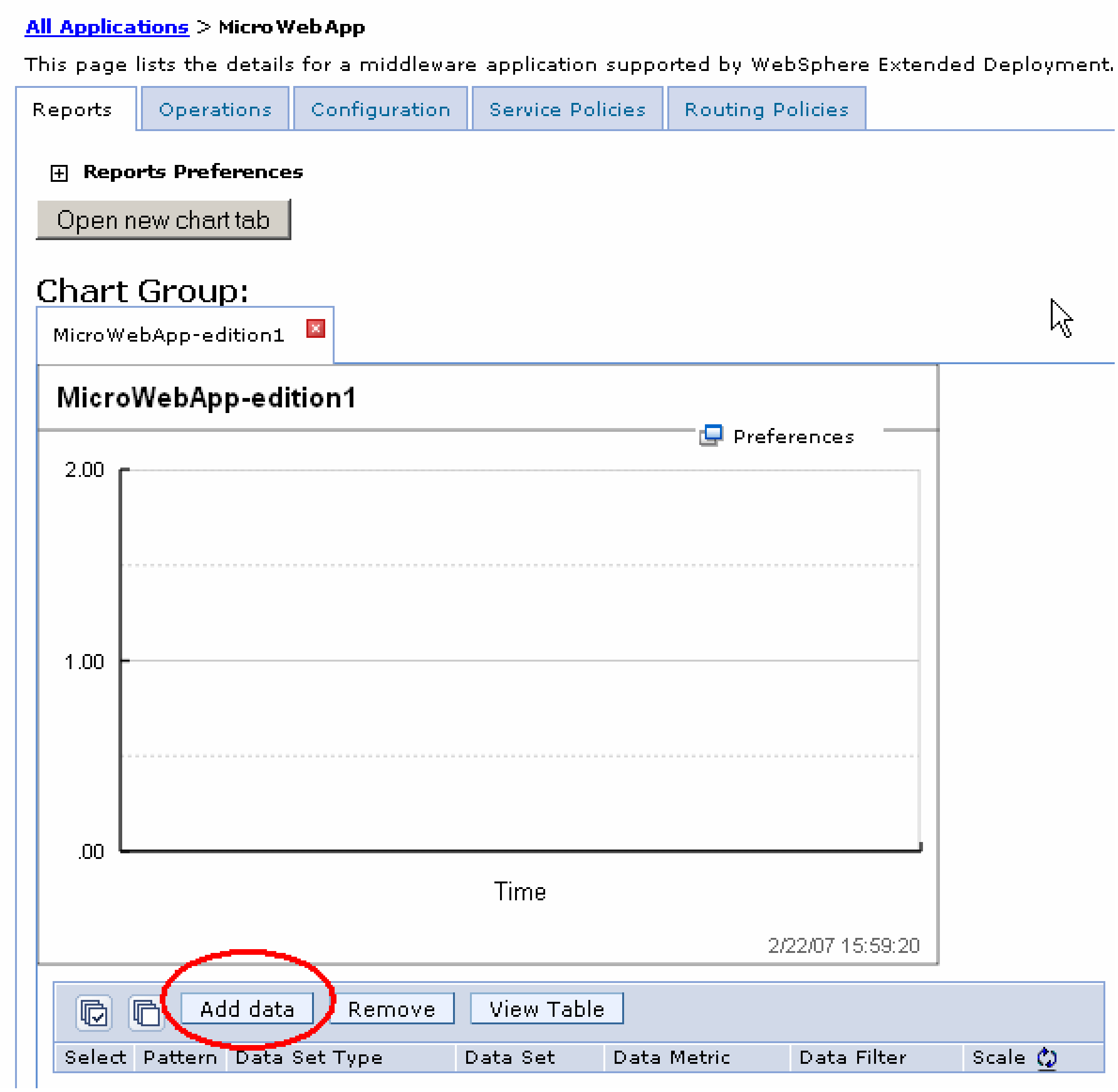
NOTE: Adobe Acrobat Reader 6.0 contains a back-level SVG viewer that will cause Internet Explorer to fail. If this happens you can browse to http://www.adobe.com/svg and install the current SVG viewer.



All Applications > MicroWebApp

This page lists the details for a middleware application supported by WebSphere Extended Deployment.

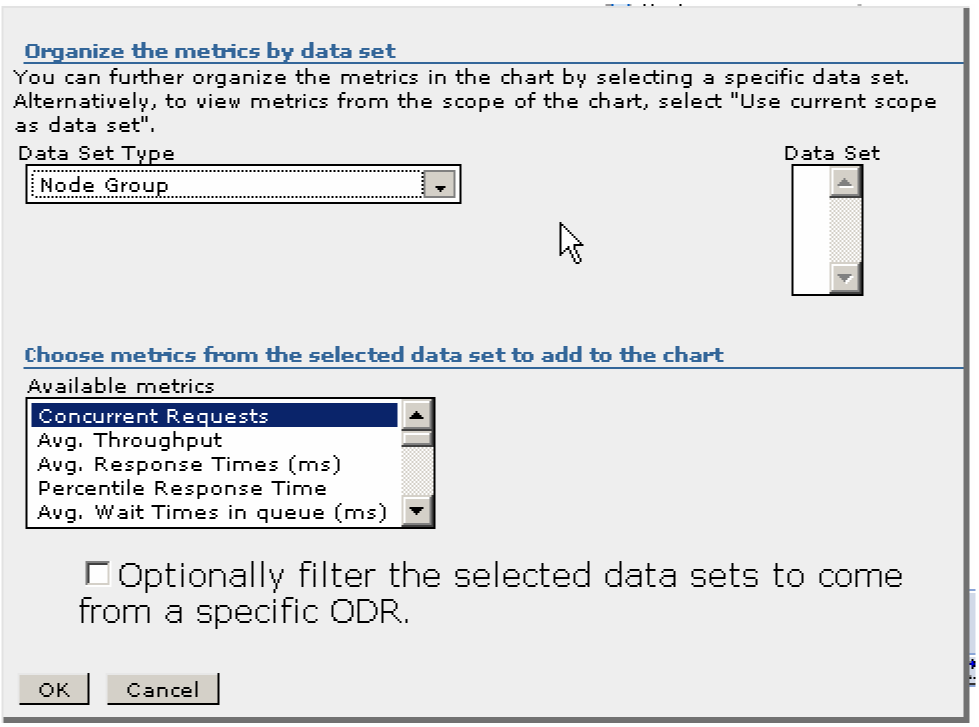
__ a. The Reports pane will draw but no data will be graphed.



All Applications > Micro Web App

This page lists the details for a middleware application supported by WebSphere Extended Deployment.

__ b. Click on **Add data** just below the chart for MicroWebApp. A gray pop-up should appear with pulldown and selection capabilities for **Data Set Type**, **Data Set** and **Available metrics.**



Organize the metrics by data set

You can further organize the metrics in the chart by selecting a specific data set. Alternatively, to view metrics from the scope of the chart, select "Use current scope as data set".

Data Set Type: Node Group

Data Set

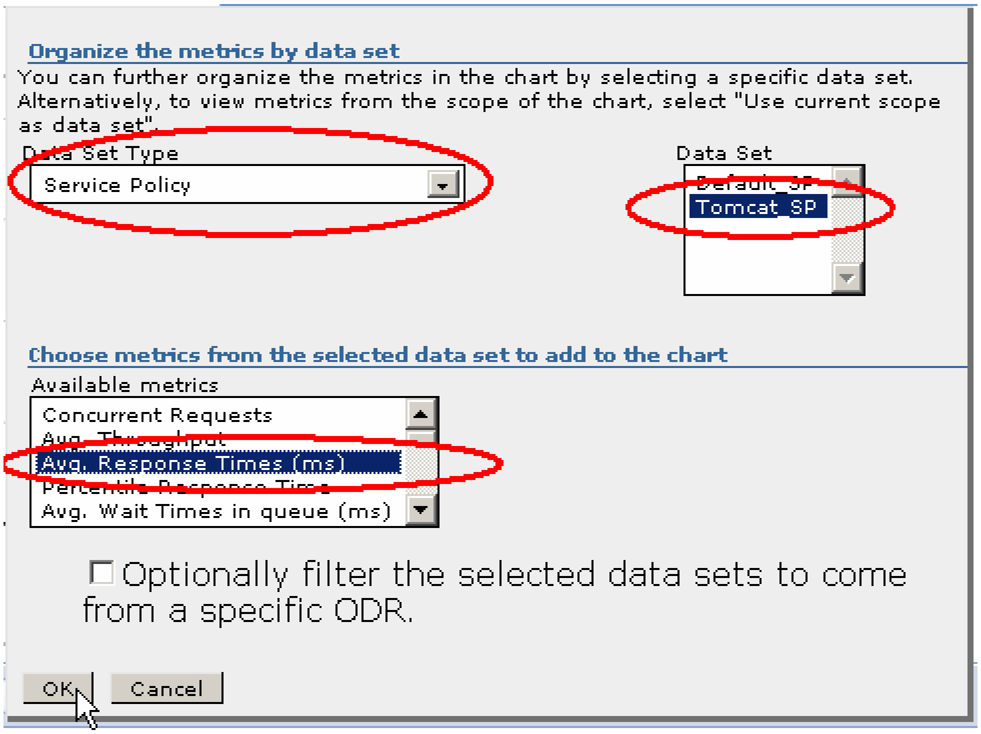
Choose metrics from the selected data set to add to the chart

Available metrics

- Concurrent Requests
- Avg. Throughput
- Avg. Response Times (ms)
- Percentile Response Time
- Avg. Wait Times in queue (ms)

☐ Optionally filter the selected data sets to come from a specific ODR.

__ c. Select **Data Set Type** of ***Service Policy*** (anticipate a brief pause), then select **Data Set** of ***Tomcat_SP***, and **Available metrics** of ***Avg. Response Times (ms)***. Then click **OK.**



Organize the metrics by data set

You can further organize the metrics in the chart by selecting a specific data set. Alternatively, to view metrics from the scope of the chart, select "Use current scope as data set".

Data Set Type: Service Policy

Data Set:

- Default_SP
- Tomcat_SP

Choose metrics from the selected data set to add to the chart

Available metrics

- Concurrent Requests
- Avg. Throughput
- Avg. Response Times (ms)
- Percentile Response Time
- Avg. Wait Times in queue (ms)

☐ Optionally filter the selected data sets to come from a specific ODR.

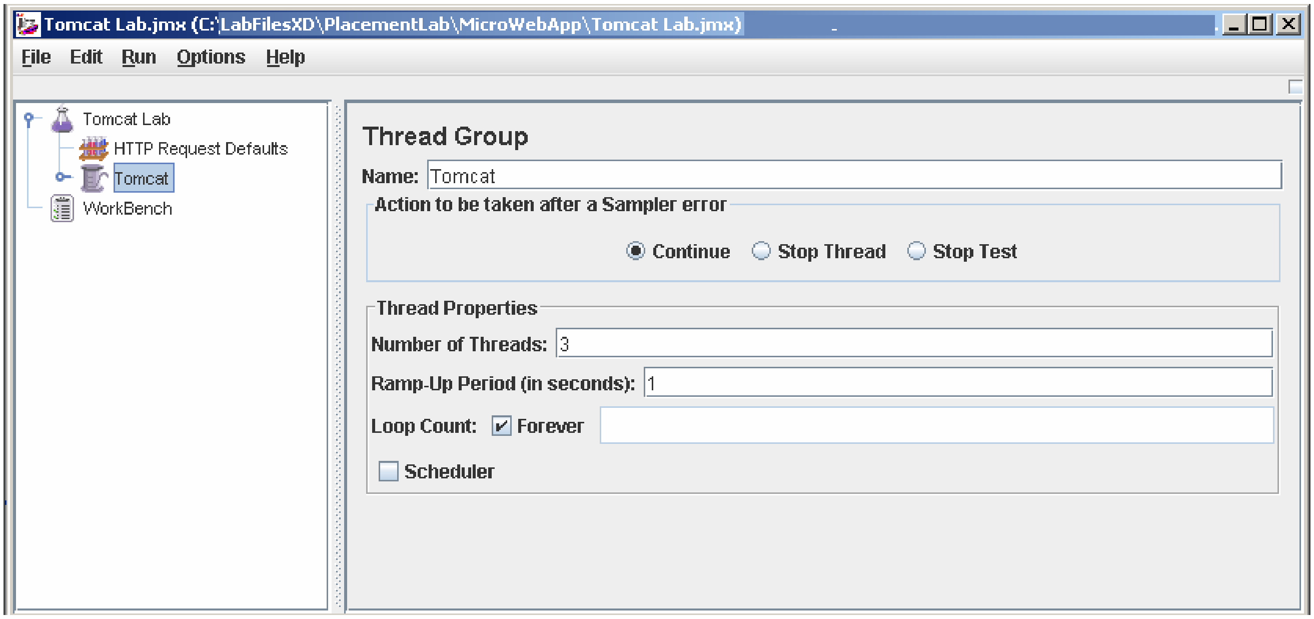
Note: In this lab you will use Apache jMeter to generate a simulated load on the MicroWebApp application. This tool is freely available from http://jakarta.apache.org/jmeter/

____ 43. Start jMeter on hostA, the machine running the Deployment Manager and the ODR.

__ a. Open a Command Prompt and navigate to C:\LabFilesXD\PlacementLab.

__ b. Type **jmeter MicroWebApp\Tomcat Lab.jmx** and press **enter.** The necessary jars will be loaded to run the test tool. Once that is complete, a GUI will appear that allows you to run the test tool.

__ c. Expand the Tomcat Lab test plan. This plan will drive a light load against your Web module installed on the Tomcat server.


Tomcat Lab.jmx (C:\LabFilesXD\PlacementLab\MicroWebApp\Tomcat Lab.jmx)
File
Edit
Run
Options
Help
Tomcat Lab
HTTP Request Defaults
Tomcat
WorkBench

Thread Group

Name: Tomcat

Action to be taken after a Sampler error

● Continue ○ Stop Thread ○ Stop Test

Thread Properties

Number of Threads: 3

Ramp-Up Period (in seconds): 1

Loop Count: ☑ Forever

□ Scheduler

__ d. Select **Start** from the **Run** menu to drive the stress tool.

____ 44. Observe the application average response times in the Reports tab of MicroWebApp.

____ 45. Stop the stress tool and add additional sessions within JMeter.

__ a. Select **Stop** from the **Run** menu in the stress tool.

__ b. Check back in Reports tab in the administrative console. You should see a drop in the average response time to zero.

__ c. In the JMeter interface, change the number of threads to **20**.

__ d. Restart the load by selecting **Start** from the **Run** menu.

____ 46. Run that load stress for about 2 minutes

____ 47. In the administration console, expand **Servers / Other middleware servers** and click on **Apache Tomcat server.** Check to see if only 1 server is still running. At some higher load point, the Application Placement Controller should start the second Apache Tomcat server.

____ 48. If the second Tomcat server has not started, then stop the stress tool and add additional 20 sessions within JMeter, and restart JMeter (**Run / Stop,** increase **number of threads, Run / Start)**. Wait another 2 minutes, and then check the status of the Apache Tomcat servers again. Repeat this process every 2 minutes, adding 20 sessions each time. Once the number of sessions reaches 80, the second Apache Tomcat server should have started.

NOTE: The amount of stress that is required to fully utilize your node is dependent upon your environment (for example, RAM and processor speed).

Intended results: as load increases and the processor utilization for hostB eventually exceeds 90 per cent for a sustained period over 2 minutes, the Application Placement Controller will start the second instance of the hostC_Tomcat server.

Non WebSphere dynamic operations

____ 49. Stop the Stress tool

__ a. Select **Stop** from the **Run** menu.

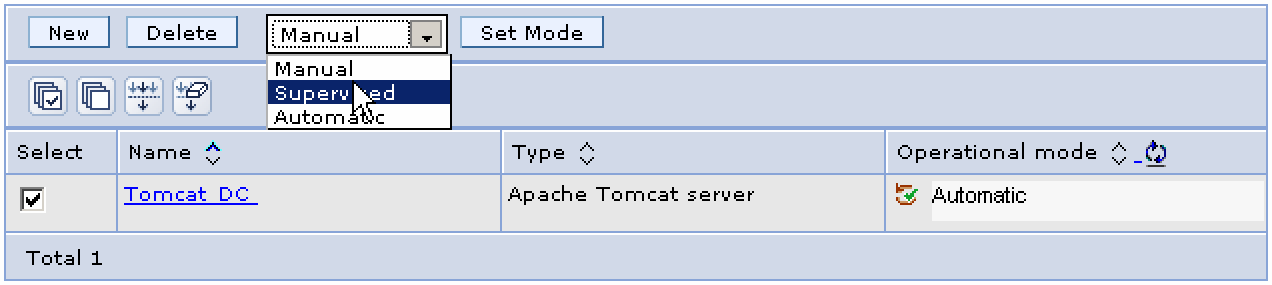
____ 50. Set the dynamic clusters' Operational Mode to Supervised in the administrative console

__ a. Expand **Servers**.

__ b. Click **Dynamic Clusters**.

__ c. Select the Tomcat_DC dynamic cluster

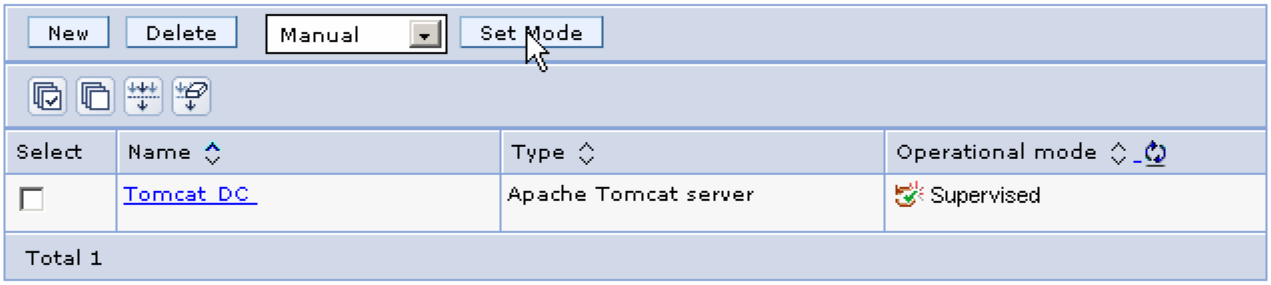
__ d. Select **Supervised** in the drop-down list.



Select	Name	Type	Operational mode
☑	Tomcat_DC	Apache Tomcat server	Automatic

Total 1

__ e. Click **Set mode**.



Select	Name	Type	Operational mode
□	Tomcat_DC	Apache Tomcat server	Supervised

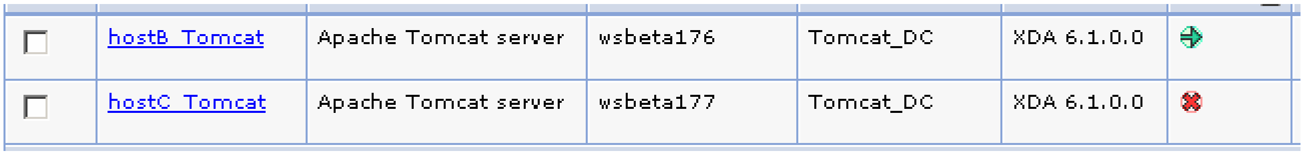
Total 1

____ 51. Manually set initial conditions again.

__ a. Expand **Servers / Other middleware servers,** and click on **Apache Tomcat servers.**

__ b. Issue the appropriate start and stop commands so that hostB_Tomcat is started and hostC_Tomcat is stopped.

Non WAS Dynamic Operations



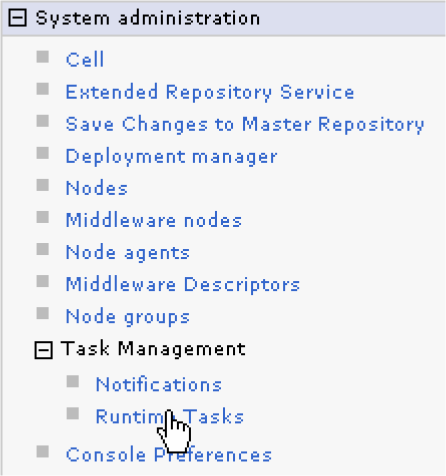
☐	hostB_Tomcat	Apache Tomcat server	wsbeta176	Tomcat_DC	XDA 6.1.0.0	➔
☐	hostC_Tomcat	Apache Tomcat server	wsbeta177	Tomcat_DC	XDA 6.1.0.0	✖

____ 52. Start the stress again.

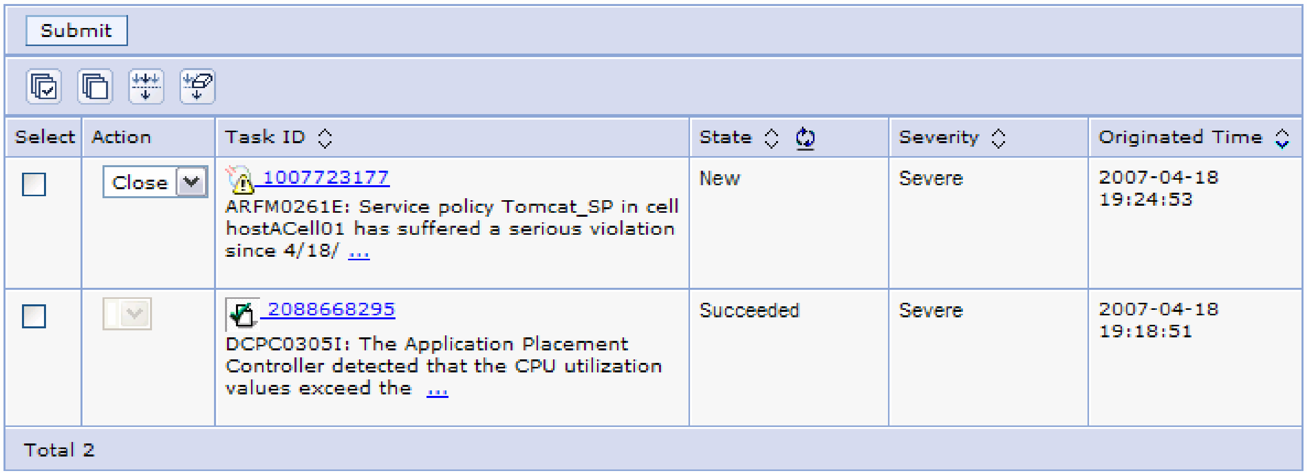
__ a. In JMeter, select **Start** from the **Run** menu to start the stress.

____ 53. Review the runtime tasks to see what tasks are suggested by Extended Deployment.

__ a. Expand **System administration / Task Management** , and click on **Runtime Tasks**.



__ b. After the stress has run for a while, a task will appear, or even two or more. Two common possibilities are that you can receive a task asking for permission to start an additional Tomcat server because of high processor utilization, or you will receive a notification of a severe violation of the Tomcat_SP service policy.



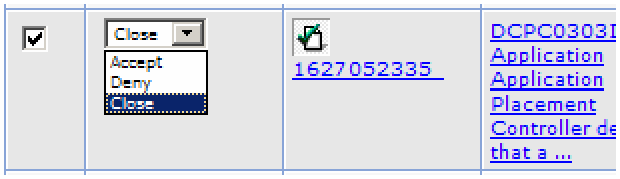
Select	Action	Task ID	State	Severity	Originated Time
☐	Close	1007723177 ARFM0261E: Service policy Tomcat_SP in cell hostACell01 has suffered a serious violation since 4/18/ ...	New	Severe	2007-04-18 19:24:53
☐		2088668295 DCPC0305I: The Application Placement Controller detected that the CPU utilization values exceed the ...	Succeeded	Severe	2007-04-18 19:18:51

Total 2

In Supervised mode you must manually determine whether to accept or deny these tasks. This is different from the Automatic operational mode, which allows Extended Deployment to run the tasks on its own. If you do not take action for some time the task will expire, not allowing the runtime task to run.

__ c. Select the **Task Explanation** to better understand the suggested action. This will open a new window with the task information.

__ d. Disable the runtime task. Select **Close** in the drop-down list. Select the check box beside the runtime task. Press the **Execute** button. Selecting **Close** tells the system that the operator will manually take the recommended action. Had you selected **Accept** the system would have taken the action on your behalf. Selecting **Deny** tells the system you do not want the task to occur.



☑	Close Accept Deny Close	1627052335	DCPC0303I Application Application Placement Controller de that a ...

____ 54. Stop the stress by selecting **Stop** from the **Run** menu in JMeter.

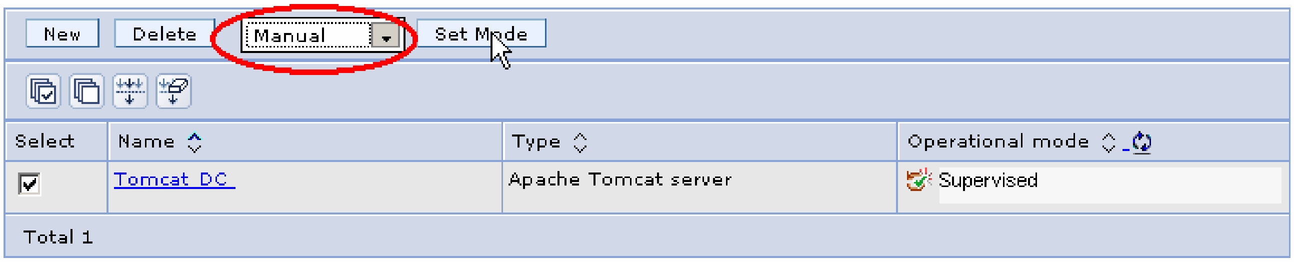
____ 55. Set the dynamic clusters' Operational Mode to Manual.

__ a. In the Navigation panel, expand **Servers** and click **Dynamic Clusters**.

__ b. Select all of the dynamic clusters in the table.

__ c. Select **Manual** in the drop-down list and click **Set Mode**.

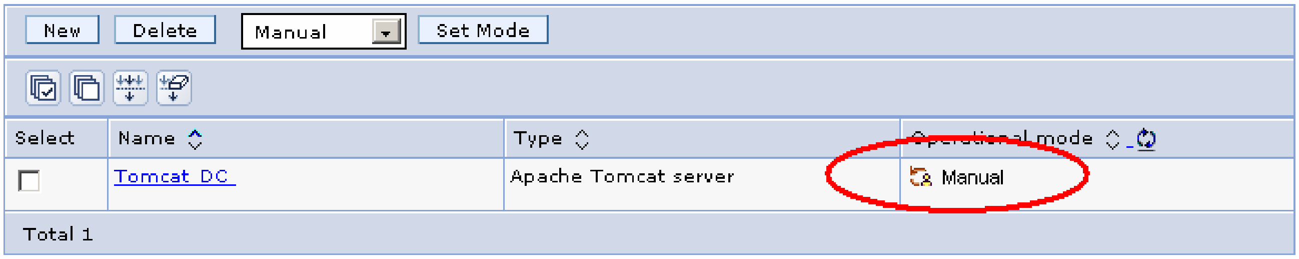
_



Manual

Select	Name	Type	Operational mode
☑	Tomcat DC	Apache Tomcat server	Supervised

Total 1



Manual

Select	Name	Type	Operational mode
☐	Tomcat DC	Apache Tomcat server	Manual

Total 1

____ 56. Stop the servers.

__ a. In the Navigation panel, expand **Servers** and select **Application Servers**.

__ b. Select all of the running dynamic cluster instances.

__ c. Click **Stop**.

__ d. Wait for confirmation that the servers have been stopped.

What you did in this exercise

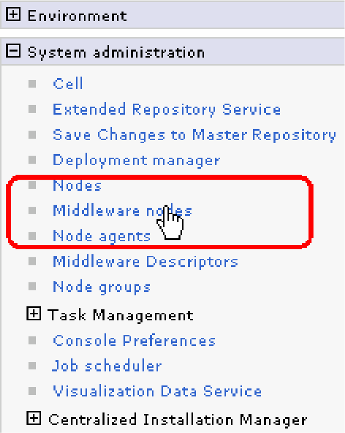
In this exercise, you configured WebSphere Extended Deployment for Application Server Placement for non-WebSphere servers. You learned how to create the agents on your systems that control the Tomcat servers. You defined Tomcat servers and a dynamic cluster to encapsulate the Tomcat servers. You learned how to register a Middleware application that is associated with the Tomcat dynamic cluster. You then created the service policy and transaction class for your business goals. The Reports features in the product provides an easy and quick method to review the individual services, application performance statistics, and the overall WebSphere cell. With a stress tool you were able to observe how Extended Deployment reacts to the different loads to meet the service goals that you defined.

Non WebSphere dynamic operations

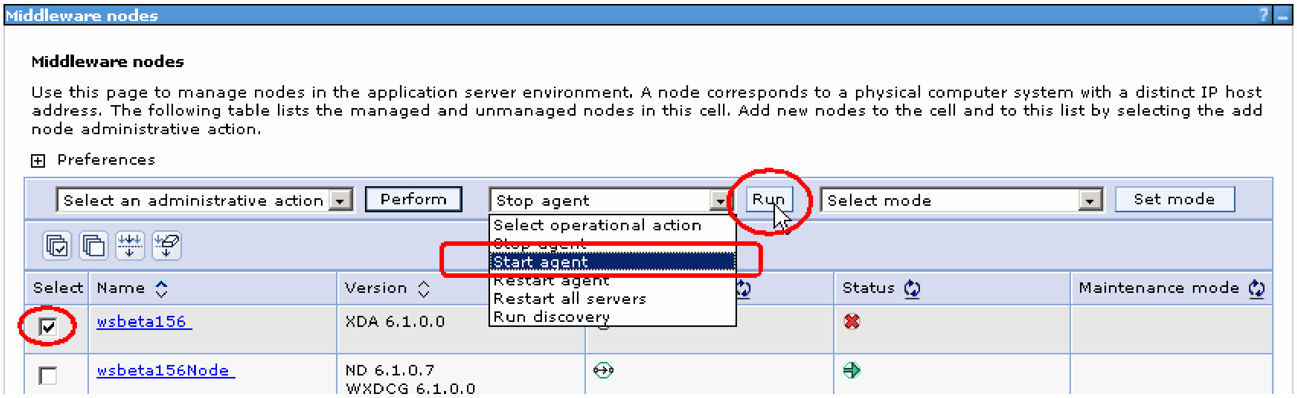
Appendix A – Starting a middleware node from the administrative console

The “Lab Requirements” section of this lab shows a multi-machine environment that is required to complete the exercise. This section will walk you through creating such an environment, provided that you have already completed the installation lab exercise.

____ 1. In the administrative console, expand **System administration,** and then click on **Middleware nodes.**



____ 2. Select the middleware node you want to start. Then in the “Select operational action” pulldown, select **Start agent**, then click **Run.**


Middleware nodes

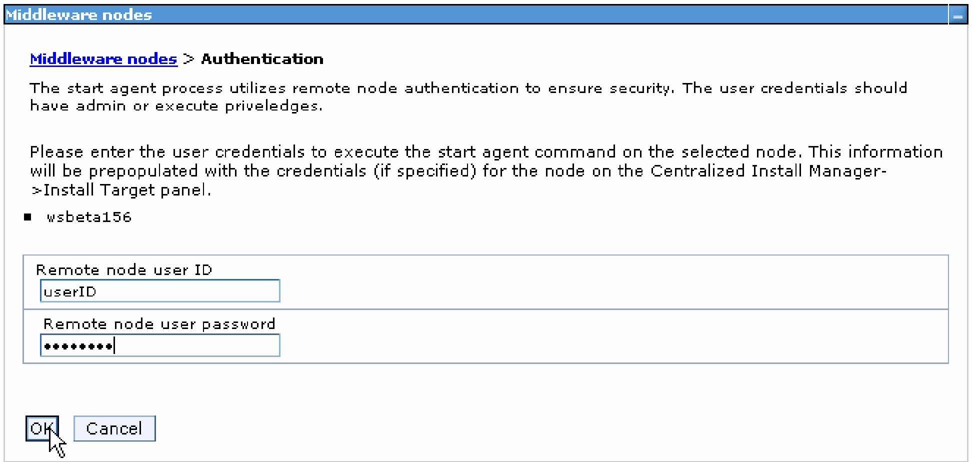
Middleware nodes

Use this page to manage nodes in the application server environment. A node corresponds to a physical computer system with a distinct IP host address. The following table lists the managed and unmanaged nodes in this cell. Add new nodes to the cell and to this list by selecting the add node administrative action.

⊞ Preferences

Select	Name	Version		Status	Maintenance mode
☑	wsbeta156	XDA 6.1.0.0		✖	
☐	wsbeta156Node	ND 6.1.0.7 WXDCG 6.1.0.0	⇔	➔	

____ 3. Type in the values for **Remote node user ID** and **Remote node user password.** Then click **OK.**



Middleware nodes

Middleware nodes > **Authentication**

The start agent process utilizes remote node authentication to ensure security. The user credentials should have admin or execute priveledges.

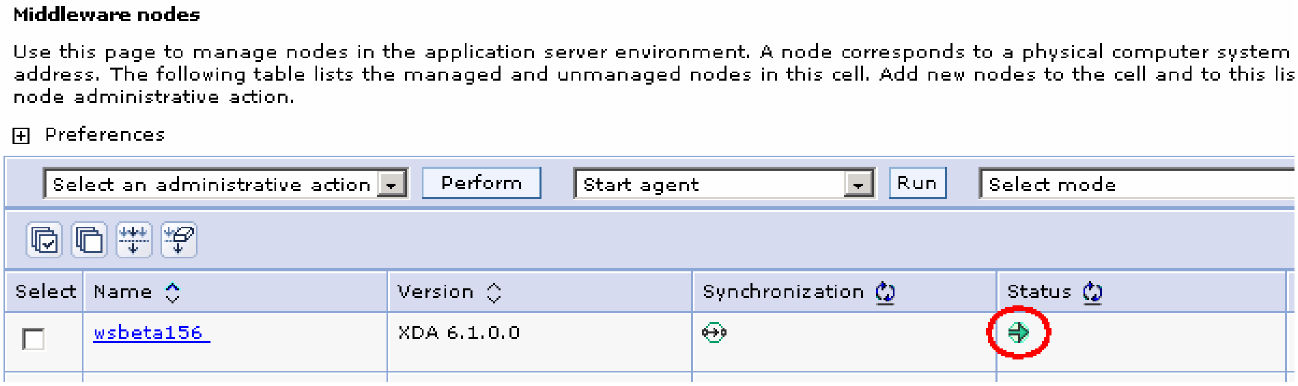
Please enter the user credentials to execute the start agent command on the selected node. This information will be prepopulated with the credentials (if specified) for the node on the Centralized Install Manager->Install Target panel.

- wsbeta156

Remote node user ID: userID

Remote node user password: ••••••••

_____ 4. The list will display and the Status field should indicate "Started" ➔. If it displays as still "Stopped", click the Refresh icon ↻.



Middleware nodes

Use this page to manage nodes in the application server environment. A node corresponds to a physical computer system address. The following table lists the managed and unmanaged nodes in this cell. Add new nodes to the cell and to this lis node administrative action.

Select	Name	Version	Synchronization	Status
☐	wsbeta156	XDA 6.1.0.0	⇔	➔

Non WebSphere dynamic operations

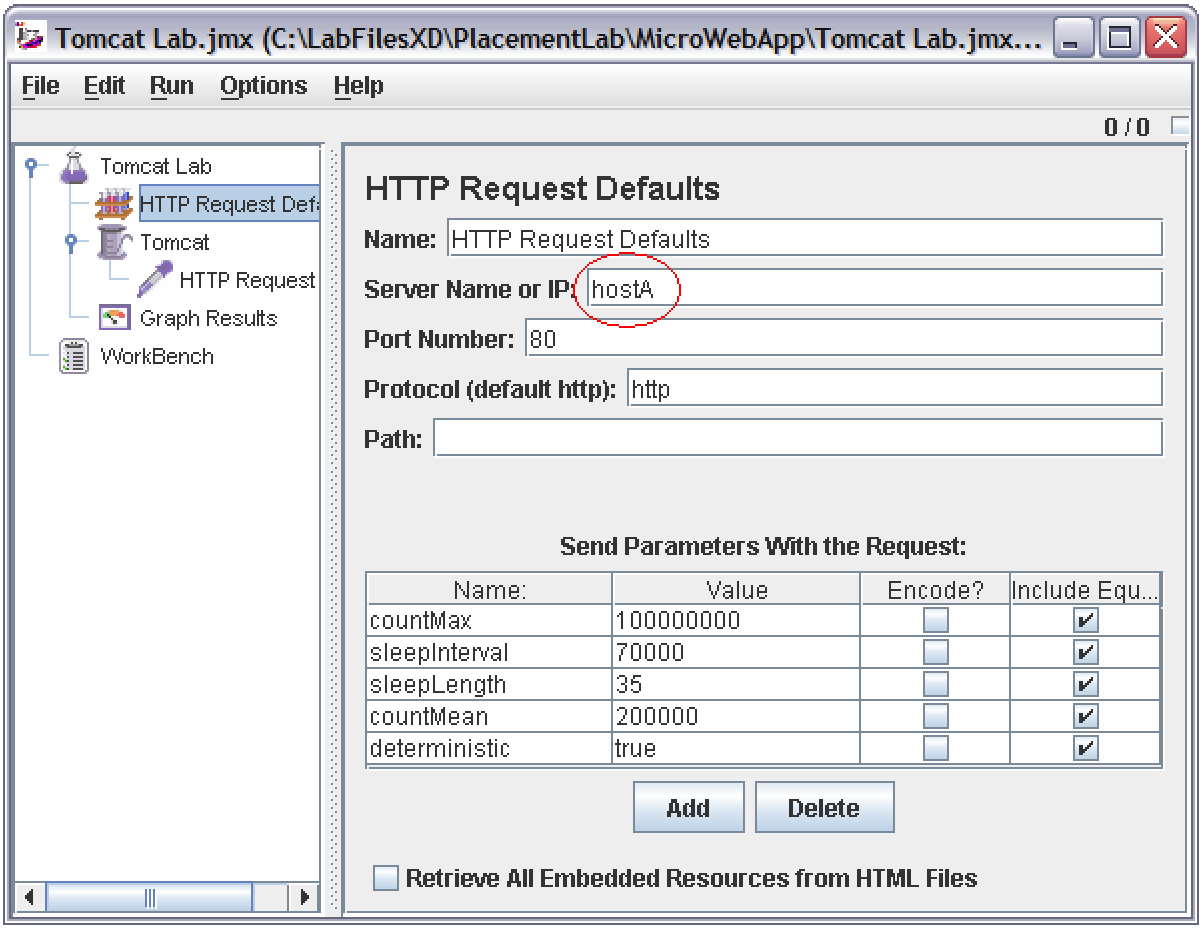
Appendix B – Customizing Load Generator

The lab instructions use Apache jMeter to generate a simulated load on the MicroWebApp application. This tool is freely available from http://jakarta.apache.org/jmeter/

However, you are free to use your favorite load generator. This is the URL used in the classroom:

- http://hostA/tomcat/CpuAndSleepBound?countMax=100000000&
  sleepInterval=70000&sleepLength=35&countMean=200000&deterministic=true

If you use the included JMeter test plan, you may need to change the host name to match your configuration, first select “HTTP Request Defaults” in the test plan and change “Server Name or IP” to the host name for the server hosting the on-demand router.



HTTP Request Defaults

Name: HTTP Request Defaults

Server Name or IP: hostA

Port Number: 80

Protocol (default http): http

Path:

Send Parameters With the Request:

Name:	Value	Encode?	Include Equ...
countMax	100000000	□	☑
sleepInterval	70000	□	☑
sleepLength	35	□	☑
countMean	200000	□	☑
deterministic	true	□	☑

□ Retrieve All Embedded Resources from HTML Files