

Estimated time 2:00

## Compute grid

What this exercise is about .....	1
Lab requirements .....	2
What you should be able to do .....	2
Introduction .....	2
Exercise Instructions .....	3
Part 1: Lab preparation .....	4
Part 2: Configure long-running scheduler .....	5
Part 3: Set up grid execution environment.....	13
Part 4: Installing the sample grid utility applications .....	20
Install SimpleCI.ear .....	20
Install tryit.class.....	26
Install MandelbrotCI.ear .....	29
Install PostingsSample.ear .....	36
What you did in this exercise .....	47

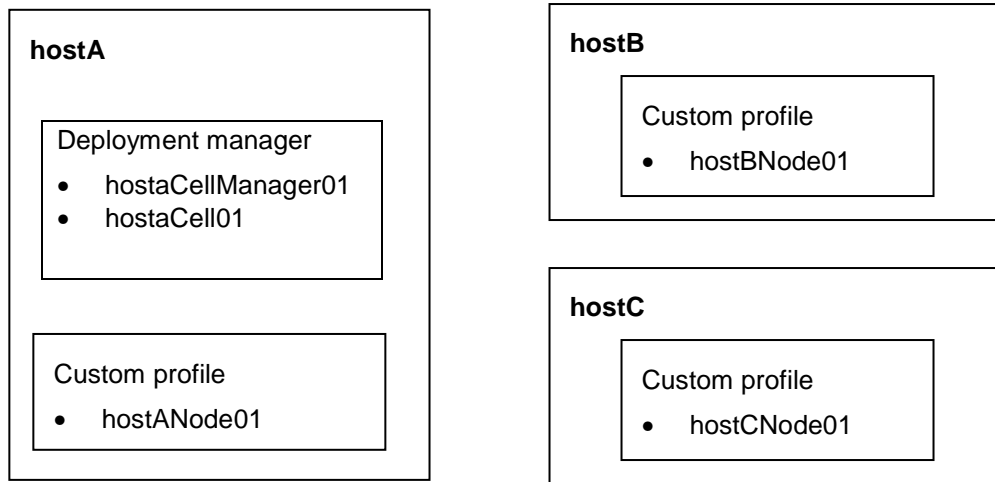
---

### What this exercise is about

The objective of this lab is to familiarize you with the steps necessary to provide you with an understanding of the grid utility application support included in WebSphere Extended Deployment V6.1.

## Lab requirements

This lab assumes that this setup is complete before starting the lab:



- The lab requires three machines: hostA, hostB, hostC.
- Deployment manager installed on hostA
- hostA also runs the long running scheduler.
- hostB and hostC are used to run two Application Server Nodes.

## What you should be able to do

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

- Configure WebSphere Extended Deployment for compute-intensive and batch operations.
- Submit a batch job.
- View status of submitted batch jobs.

## Introduction

The J2EE specification is primarily geared to support Web-based interactions -- relatively short-lived, isolated transactions – and does not provide any support for massively parallel (that is, grid) applications. Existing batch and grid frameworks do not support J2EE programming models. This forces developers to use completely different programming environments, styles, and sometimes even languages to support both interactive and batch applications.

The WebSphere Extended Deployment V6.1 Compute Grid supports two styles of long-running work:

**Batch** – A record-processing model in which the container provides transactions and checkpoint/restart mechanisms, and the application provides “process 1 record” logic. The WebSphere Extended Deployment V6.0 batch environment supports a J2EE-based batch

processing programming model, as well as the re-use of existing J2EE services/artifacts in “batch mode”. The WebSphere Extended Deployment V6.1 extends this capability to non-WebSphere, non-J2EE jobs.

**Compute-intensive** – Compute-intensive work requires significant amounts of processing to complete. Work is expressed as a sequence of parameterized steps to be run sequentially. The container provides thread of execution and the application provides everything else. The container has limited contact with work after it is started.

The WebSphere Extended Deployment V6.0 Compute Grid is composed of two major components:

**Job scheduler** – Accepts job submissions, assigns job ids, persists jobs in database, uses policy information to select which jobs to run where and when, and sends jobs to execution environments

**Execution environment** – Runs within an instance of a WebSphere Application Server and is responsible for doing the work. The execution environment can exist in a dynamic cluster, which allows instances to be started and stopped based on current and anticipated load.

## Exercise Instructions

Some instructions in this lab may be 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 vs. .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references, as follows:

Reference Variable	Windows Location	AIX/UNIX Location
<WAS_HOME>	C:\WebSphere\AppServer	/usr/WebSphere/AppServer /opt/WebSphere/AppServer
<LAB_FILES>	C:\LabfilesXD	/tmp/LabfilesXD
<TEMP>	C:\temp	/tmp

**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, C:\LabFilesXD\ would be replaced by C:/LabFilesXD/

**Note:** Throughout this lab, when the instructions state you should “**Save your changes**”, perform these actions:

- \_\_\_ 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**.

## Part 1: Lab preparation

**NOTE:** In V6.1, the job scheduler and grid execution environment are installed automatically as system applications. Default Derby databases are also installed and configured automatically for use by the scheduler and execution environment.

### Start the deployment manager

- \_\_\_ 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
```

### Start the remaining WebSphere processes

- \_\_\_ 2. Change directories to **C:\WebSphere\AppServer\profiles\hostANode01\bin**
  - \_\_\_ a. To start the node agent, enter the command: **startNode**
  - \_\_\_ b. Wait for the deployment manager to start. Verify that this line appears in the Command Prompt window:

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

- \_\_\_ 3. Start the node agent on hostB
  - \_\_\_ a. On **hostB**, open a command prompt
  - \_\_\_ b. Change directories to **C:\WebSphere\AppServer\profiles\hostBNode01\bin**
  - \_\_\_ c. Start the node agent by entering the command: **startNode**
  - \_\_\_ d. Wait for the node agent to start. Verify that this line appears in the command prompt window.

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

- \_\_\_ 4. Start the node agent on hostC
  - \_\_\_ a. On **hostC**, open a command prompt
  - \_\_\_ b. Change directories to **C:\WebSphere\AppServer\profiles\hostCNode01\bin**
  - \_\_\_ c. Enter this command to start the node agent : **startNode**
  - \_\_\_ d. Wait for the node agent to start. Verify that this line appears in the command prompt window.

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

## Part 2: Configure long-running scheduler

### Create a dynamic cluster to host the job scheduler

The job scheduler is a J2EE application. In order to make the job scheduler highly available, it must be deployed to a dynamic cluster within your WebSphere Extended Deployment environment. Use these steps to define a dynamic cluster for the job scheduler:

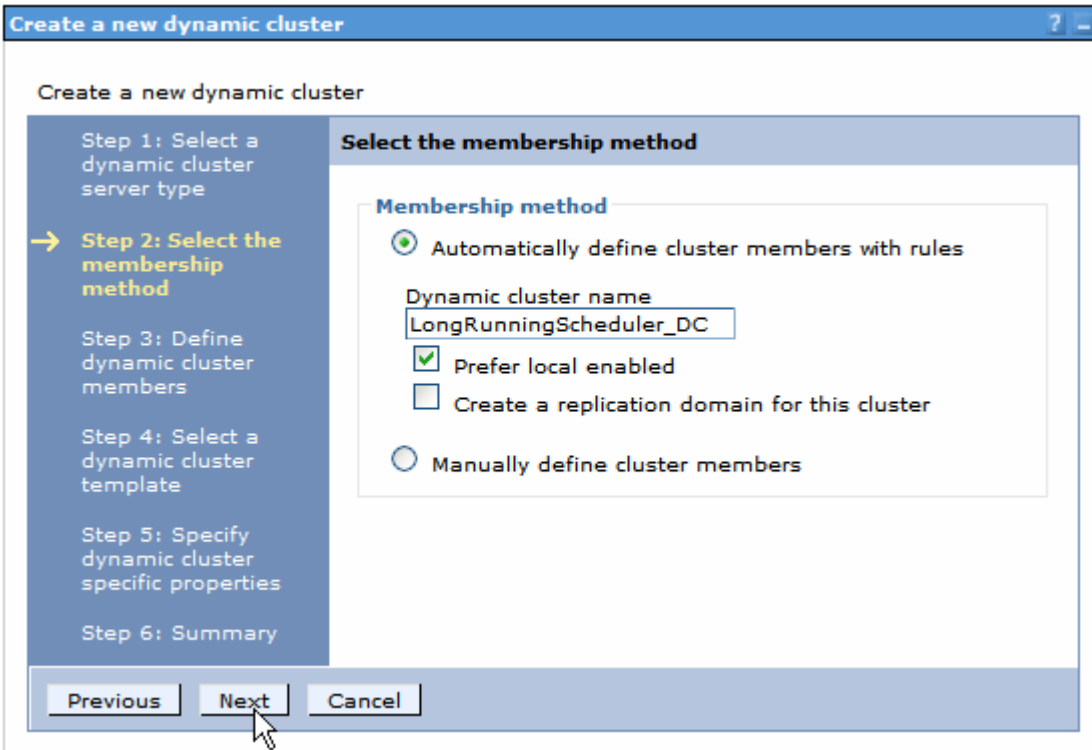
- \_\_\_ 5. Determine the nodes in your cell that should be used to host the job scheduler. The job scheduler will be accessed by the administrative console, application clients, J2EE applications and Web service requests. It will need access to the job database and the nodes that will be hosting execution environments (see below). For this lab, you will install the job scheduler on hostANode01.
- \_\_\_ 6. In the administrative console navigation tree, expand "Servers" and click on "Dynamic Clusters".
- \_\_\_ 7. Click "New".

The screenshot shows the 'Create a new dynamic cluster' wizard. The title bar reads 'Create a new dynamic cluster'. The main content area is titled 'Create a new dynamic cluster' and contains a progress bar with six steps:

- 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

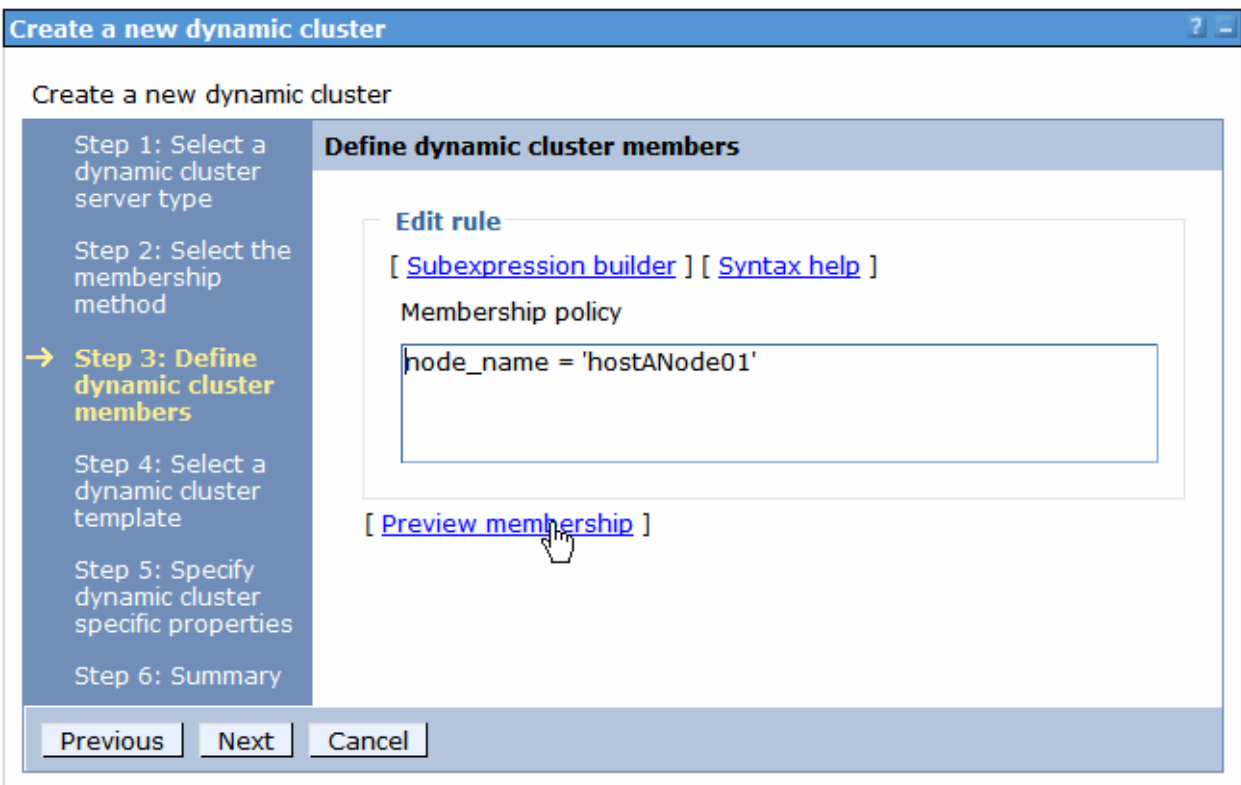
The main area is titled 'Select a dynamic cluster server type' and contains a 'Server type' dropdown menu with 'WebSphere application server' selected. At the bottom, there are 'Next' and 'Cancel' buttons.

- \_\_\_ 8. Select "WebSphere application server" (the default) and click "Next"
- \_\_\_ 9. Enter an appropriate name for the dynamic cluster, "LongRunningScheduler\_DC" for example.

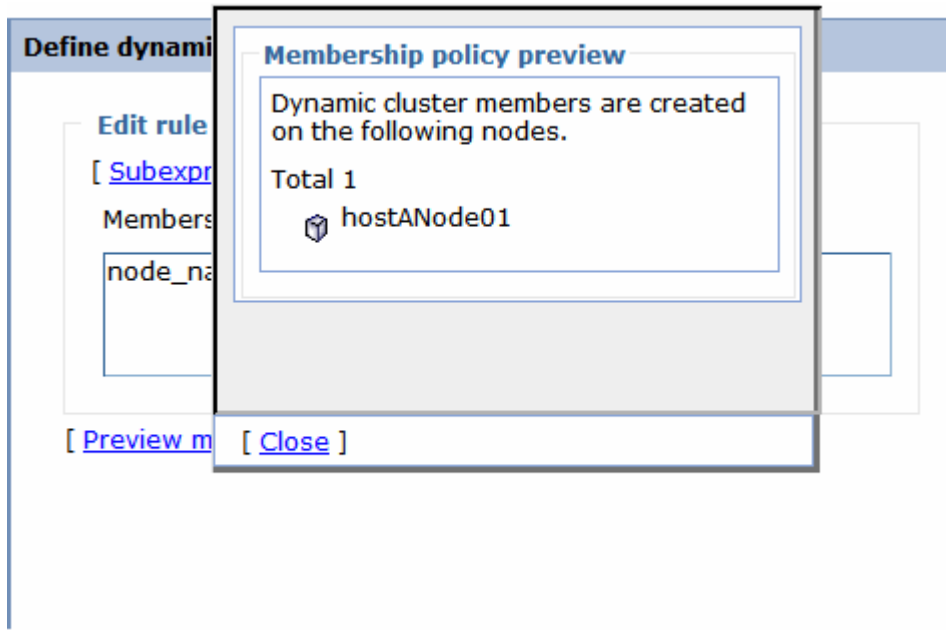


10. Click "Next"

11. Edit the "Membership policy" to include the nodes where the scheduler should run. For this lab you will install the job scheduler only on hostANode01d.



\_\_\_ 12. Click on "Preview membership" to verify you are getting the correct nodes.



\_\_\_ 13. Click "Close"

\_\_\_ 14. Click "Next"

\_\_\_ 15. Click "Next"

- \_\_\_ 16. Make sure "Keep one instance started at all times" is selected.
- \_\_\_ 17. Select "Limit the number of instances that can be started" and enter "1" in the text field.
- \_\_\_ 18. Select "No isolation requirements"

**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:  
1

Do not limit the number of instances that can start

**Vertical stacking of instances on node**

Allow more than one instance to start on the same node  
Number of instances:  
2

**Isolation preference**

**No isolation requirements**

Strict isolation

Associate with isolation group  
Isolation group name  
[ Browse ]

Previous Next Cancel

- \_\_\_ 19. click "Next"
- \_\_\_ 20. click "Finish"
- \_\_\_ 21. Save your 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**.



22. To have WebSphere Extended Deployment automatically manage the location of your job scheduler in a production environment, you would check the box next to the dynamic cluster you just created, select "Automatic" in the mode pull-down and click "Set Mode". For this lab you will leave the mode set to the default "Manual".

**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.

⊞ Preferences

New Delete Manual Set Mode

Manual  
Supervised  
Automatic

Select	Name	Type	Operational mode
<input checked="" type="checkbox"/>	<a href="#">LongRunningScheduler_DC</a>	WebSphere application server	Manual

Total 1

Enable job scheduler

As noted earlier, the job scheduler is a system application. These steps guide you through enabling it:

- \_\_\_ 23. In the navigation tree, expand "System administration" and click on "Job scheduler".
- \_\_\_ 24. In the "Scheduler hosted by" pull down, select the dynamic cluster you created for the job scheduler, "LongRunningScheduler\_DC".
- \_\_\_ 25. In the "Data source JNDI name" pull-down, select "jdbc/lrsched".

- \_\_\_ 26. Press "OK"
- \_\_\_ 27. Save your changes. Be sure to check "Synchronize changes with Nodes"

Verify the job scheduler installation

- \_\_\_ 28. Find the Application Servers' ports
  - \_\_\_ a. In the administrative console navigation tree, expand "Servers" and click "Application Servers".
  - \_\_\_ b. Select LongRunningScheduler\_DC\_hostANode01.
  - \_\_\_ c. Under "Communications", expand "Ports".

\_\_\_ d. Note the port associated with WC\_defaulthost. In this example it is 9080.

Communications	
Ports	
Port Name	Port
BOOTSTRAP_ADDRESS	9811
SOAP_CONNECTOR_ADDRESS	8884
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	9404
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	9405
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	9406
WC_adminhost	9061
WC_defaulthost	9080
DCS_UNICAST_ADDRESS	9357
WC_adminhost_secure	9044
WC_defaulthost_secure	9444
SIB_ENDPOINT_ADDRESS	7276
SIB_ENDPOINT_SECURE_ADDRESS	7286
SIB_MQ_ENDPOINT_ADDRESS	5558
SIB_MQ_ENDPOINT_SECURE_ADDRESS	5578
ORB_LISTENER_ADDRESS	0

\_\_\_ e. Record this information in the table here:

Server	Port
LongRunningScheduler_DC_hostANode01	

\_\_\_ 29. Start the WebSphere server hosting the job scheduler

\_\_\_ a. On the left panel of the administration console expand “Servers” and click on “Application servers”.

\_\_\_ b. Select check box next to “LongRunningScheduler\_DC\_hostANode01”

\_\_\_ c. Click “Start” and wait for the server to start

\_\_\_ 30. Open Internet Explorer and go to the Job Management Console. The URL should be `http://<hosta>:<WC_defaulthost>/jmc`. For example: `http://hostA:9080/jmc`

\_\_\_ 31. Login. For this lab security is not enabled, so you can use any user ID.

\_\_\_ 32. The open panel for the job management controller looks like this:

Compute Grid Job Management Console <b>Welcome nathan</b>	
<ul style="list-style-type: none"> <li>▪ <a href="#">Welcome</a></li> <li>☐ <b>Job Management</b> <ul style="list-style-type: none"> <li>▪ <a href="#">View jobs</a></li> <li>▪ <a href="#">Submit a job</a></li> </ul> </li> <li>☐ <b>Job Repository</b> <ul style="list-style-type: none"> <li>▪ <a href="#">View saved jobs</a></li> <li>▪ <a href="#">Save a job</a></li> </ul> </li> <li>☐ <b>Schedule Management</b> <ul style="list-style-type: none"> <li>▪ <a href="#">View schedules</a></li> <li>▪ <a href="#">Create a schedule</a></li> </ul> </li> </ul>	<p><b>Welcome</b></p> <hr/> <p>The job management console is a standalone Web interface for WebSphere Extended Deployment Compute Grid users to perform job operations depending on role privileges. Interaction with the job repository is also possible, where jobs can be saved, removed or get submitted to the repository. This console provides controlled access when security is enabled. Only authorized users who are granted the Irsubmitter and/or Iradmin roles through the WebSphere Extended Deployment administrative console can be allowed access to the job management console.</p>

---

## Part 3: Set up grid execution environment

Grid utility applications are packaged and deployed as J2EE applications that run inside of WebSphere. Except for the special steps described here, deployment of a grid utility application is analogous to deployment of a transactional J2EE application.

Grid utility applications are hosted in a grid execution environment. WebSphere Extended Deployment provides three execution environments: one for compute-intensive applications, one for transactional batch applications, and another for non-WebSphere programs. The logic for these environments is packaged as a single J2EE application (GEE.ear). This J2EE application is automatically deployed to a dynamic cluster when a grid application is installed.

These steps describe what you need to do to successfully deploy your grid utility application in a WebSphere Extended Deployment server.

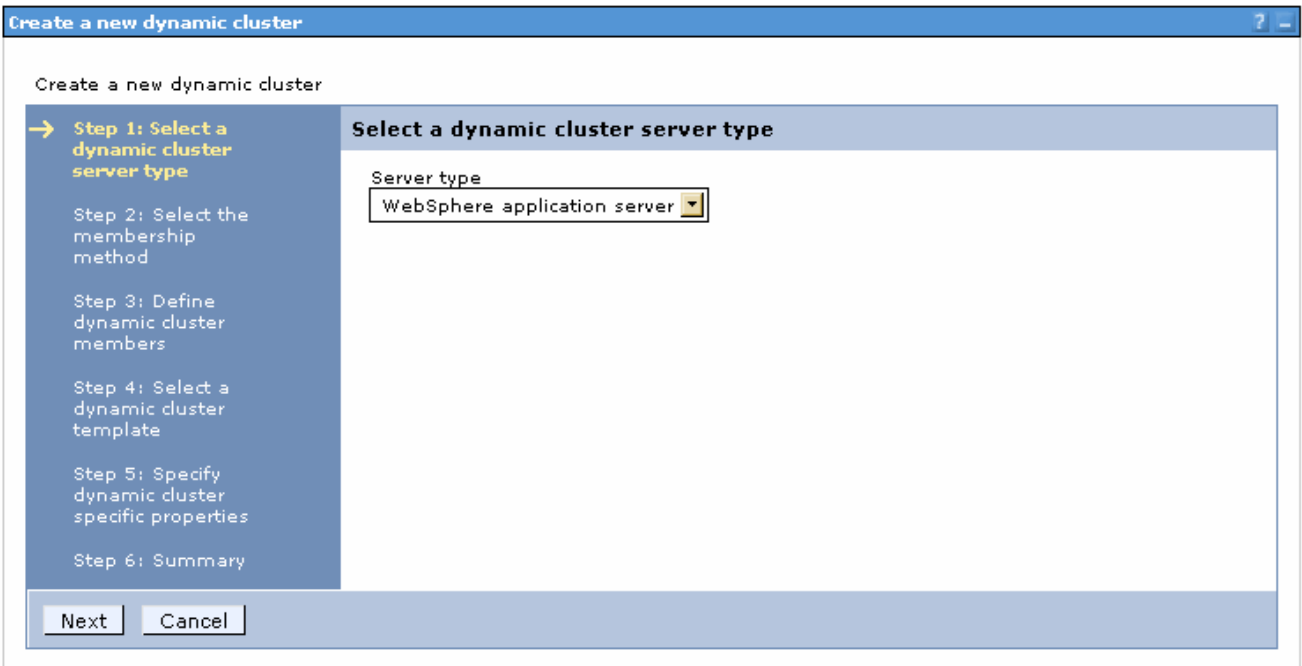
### Create a dynamic cluster for the execution environment and grid utility applications

Execution environments and grid utility applications are deployed to dynamic clusters just like transactional applications. These steps describe how to create a dynamic cluster for the execution environment and grid utility applications.

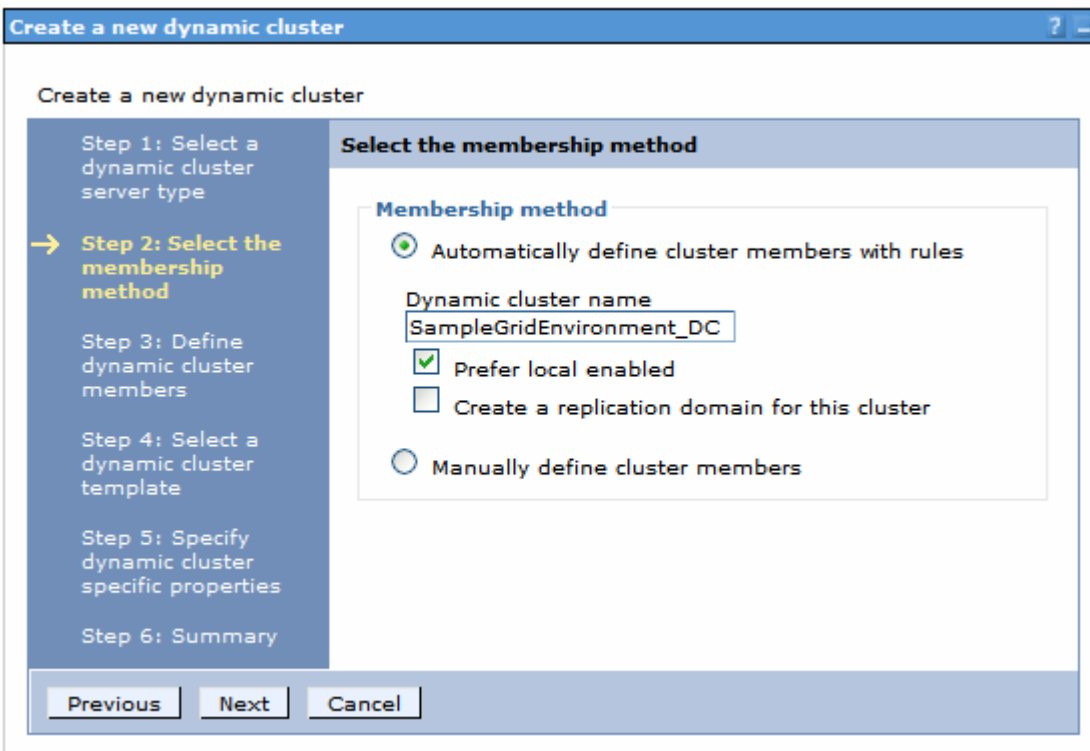
You use dynamic cluster group membership policy to tell WebSphere Extended Deployment which nodes in your cell are capable of running grid utility applications. This lab will create a dynamic cluster to host the grid application. If you already have a dynamic cluster that contains the appropriate nodes for the grid utility application, you can skip these steps.

To define a dynamic cluster for the grid execution environment:

- \_\_\_ 33. Determine the nodes in your cell that should be used to host the execution environment and grid utility application. For this lab, use **hostBNode01** and **hostCNode01**.
- \_\_\_ 34. In the administrative console navigation tree, expand "Servers" and click on "Dynamic Clusters".
- \_\_\_ 35. Click "New".



- \_\_\_ 36. Select "WebSphere application server" and click "Next"
- \_\_\_ 37. Enter an appropriate name for the dynamic cluster, "SampleGridEnvironment\_DC" for example.



- \_\_\_ 38. Click "Next"
- \_\_\_ 39. Edit the "Membership policy" to include the nodes that will host the grid applications. For this lab, use **hostBNode01** and **hostCNode01**. WebSphere Extended Deployment provides several

mechanisms to specify multiple nodes in the membership policy. One is shown in the image below. Note that node names are case sensitive.

The screenshot shows a web browser window titled "Create a new dynamic cluster". The main content area is divided into a left sidebar and a main panel. The sidebar lists six steps: Step 1: Select a dynamic cluster server type; Step 2: Select the membership method; Step 3: Define dynamic cluster members (highlighted with a yellow arrow); Step 4: Select a dynamic cluster template; Step 5: Specify dynamic cluster specific properties; Step 6: Summary. The main panel is titled "Define dynamic cluster members" and contains an "Edit rule" section with a text area for the membership policy. The policy text is "node\_name IN ('hostBNode01','hostCNode01')". Below the text area is a "[ Preview membership ]" link. At the bottom of the main panel are three buttons: "Previous", "Next", and "Cancel".

**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**

**Edit rule**

[ [Subexpression builder](#) ] [ [Syntax help](#) ]

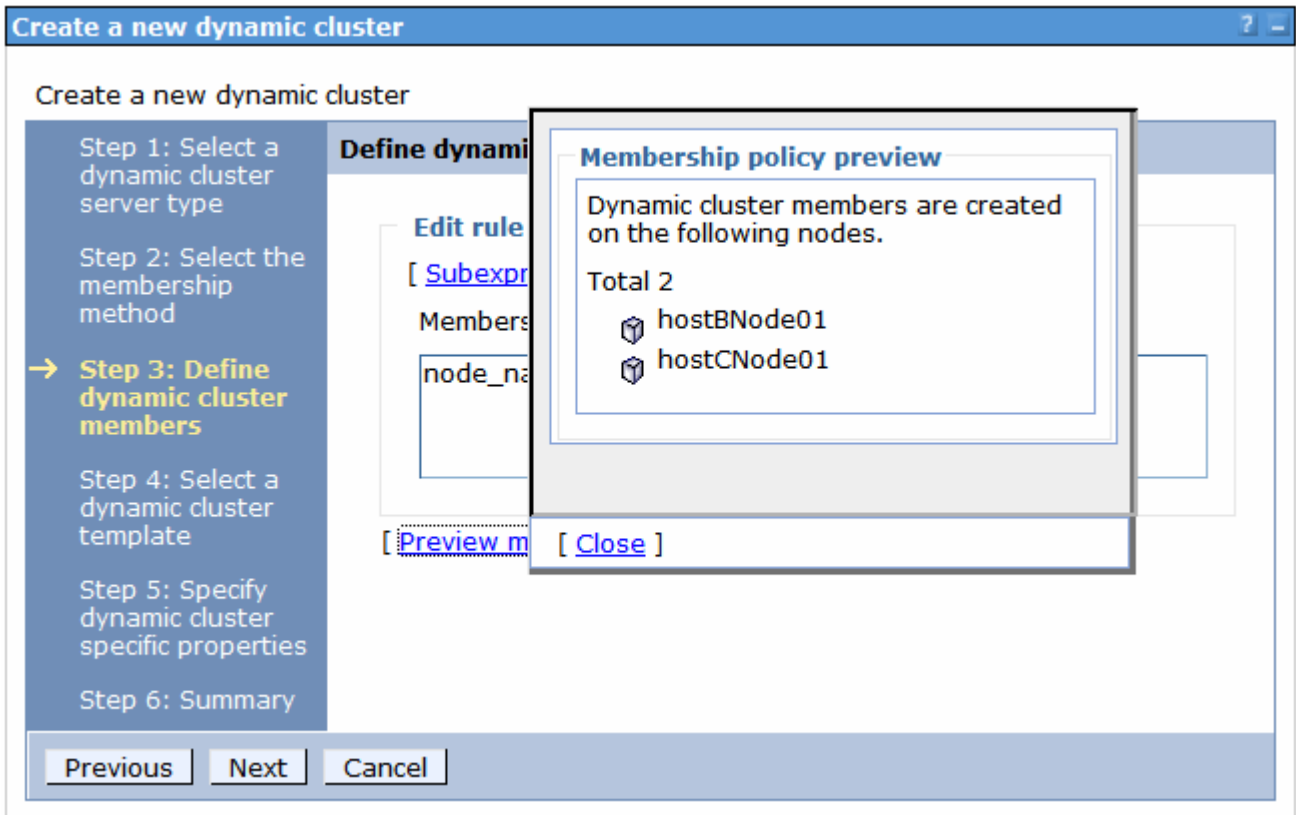
Membership policy

```
node_name IN ('hostBNode01','hostCNode01')
```

[ [Preview membership](#) ]

[Previous](#) [Next](#) [Cancel](#)

\_\_\_ 40. Click on "Preview membership" to verify you are getting the correct node group.



\_\_\_ 41. Click "Close"

\_\_\_ 42. Click "Next"

\_\_\_ 43. Click "Next"



- \_\_\_ 44. Select "Keep one instances started at all times".
- \_\_\_ 45. Make sure the default "Do not limit the number of instances that can start" is selected.
- \_\_\_ 46. Select "No isolation requirements"

- \_\_\_ 47. click "Next"
- \_\_\_ 48. click "Finish"
- \_\_\_ 49. Save your changes. Be sure to check "Synchronize changes with Nodes"

Install the execution environment

The execution environment deploys automatically when you install a grid utility application.

- \_\_\_ a. In the administrative console navigation panel, expand **Servers** and click **Application Servers**.

\_\_ b. Select any instances of SampleGridEnvironment\_DC\_hostXNodeN that are running.

Select	Name	Node	Version	Cluster Name	Status
<input type="checkbox"/>	<a href="#">LongRunningScheduler_DC_hostANode01</a>	hostANode01	ND 6.1.0.3 XD 6.1.0.0	LongRunningScheduler_DC	➔
<input checked="" type="checkbox"/>	<a href="#">SampleGridEnvironment_DC_hostBNode01</a>	hostBNode01	ND 6.1.0.3 XD 6.1.0.0	SampleGridEnvironment_DC	➔
<input checked="" type="checkbox"/>	<a href="#">SampleGridEnvironment_DC_hostCNode01</a>	hostCNode01	ND 6.1.0.3 XD 6.1.0.0	SampleGridEnvironment_DC	➔
Total 3					

\_\_ c. Click **Stop**.

\_\_ d. Wait for confirmation that the servers are stopped.

\_\_\_ 50. The grid execution environment (GEE) exposes a Web services interface to receive job information from the scheduler. By default, the GEE Web services module is bound to the grid\_host virtual host of the servers where the GEE application is installed. To see this in the administrative console, expand Environment and click on “Virtual Hosts” .

Select	Name
<input type="checkbox"/>	<a href="#">admin_host</a>
<input type="checkbox"/>	<a href="#">default_host</a>
<input type="checkbox"/>	<a href="#">grid_host</a>
<input type="checkbox"/>	<a href="#">proxy_host</a>
Total 4	

51. To have WebSphere Extended Deployment automatically manage the location of your grid batch environment in a production environment, expand “Servers” in the administrative console and click on “Dynamic clusters”. Check the box next to the dynamic cluster you just created, Select “Automatic” in the mode pull-down, and click “Set mode”. For this lab you will leave mode set to the default “Manual.”

**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.

⊞ Preferences

New Delete Manual Set Mode

Manual  
Supervised  
Automatic

Select	Name	Type	Operational mode
<input type="checkbox"/>	<a href="#">LongRunningScheduler_DC</a>	WebSphere application server	Manual
<input checked="" type="checkbox"/>	<a href="#">SampleGridEnvironment_DC</a>	WebSphere application server	Manual

Total 2

---

## Part 4: Installing the sample grid utility applications

It is now time to install the grid utility application itself. A grid utility application is installed exactly like a J2EE transactional application. When mapping the EJB modules of the grid utility application to servers, be sure to select the same dynamic cluster you created on page 14.

### Sample applications

WebSphere Extended Deployment includes three sample applications:

#### SimpleCI.ear

A simple compute-intensive application that performs computationally-intensive mathematical calculations for a user-specified amount of time.

#### MandelbrotCI.ear

A compute-intensive application with a Web interface that computes and renders Mandelbrot fractal images.

#### PostingsSample.ear

A transactional batch application that mimics account transaction processing.

All three sample applications are included in the <WAS\_HOME>/installableApps directory. In addition, the lab includes a fourth:

#### Tryit.class

A simple Java application to run in the grid utility.

## Install SimpleCI.ear

SimpleCI is a simple compute-intensive application that performs computationally-intensive mathematical calculations for a user-specified amount of time.

1. In the administrative console navigation tree, expand "Applications" and click on "Install New Application".

- \_\_\_ 2. Select "Remote file system", and Browse to hostACellManager01 → "C:\WebSphere\AppServer\installableApps", select "SimpleCI.ear" and click "OK".

- \_\_\_ 3. Click "Next".  
If you receive an Application Security warning, click "Continue".

- \_\_\_ 4. In “Step 1: Select installation Options”, clear the check box for “Deploy enterprise beans”

**Step 1: Select installation options**

Step 2 Map modules to servers

Step 3 Summary

**Select installation options**

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

Precompile JavaServer Pages files

Directory to install application

Distribute application

Use Binary Configuration

Deploy enterprise beans

Application name

Application edition

Edition description

Create MBeans for resources

Enable class reloading

Reload interval in seconds

- \_\_\_ 5. Press “Next”

- \_\_\_ 6. In “Step 2: Map modules to servers”, select SimpleCIEJB from the module list, and SampleGridEnvironment\_DC from the list of “Clusters and Servers”. **Click “Apply”**

**Map modules to servers**

Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are routed through.

Clusters and Servers:

- WebSphere:cell=hostACell01,cluster=SampleGridEnvironment\_DC
- WebSphere:cell=hostACell01,cluster=Tomcat\_DC
- WebSphere:cell=hostACell01,cluster=LongRunningScheduler\_DC
- WebSphere:cell=hostACell01,node=wsbeta177,server=middlewareagent
- WebSphere:cell=hostACell01,node=wsbeta177,server=hostC\_Tomcat

Select	Module	URI	Server
<input checked="" type="checkbox"/>	SimpleCIEJB	SimpleCIEJB.jar,META-INF/ejb-jar.xml	WebSphere:cell=hostACell01,cluster=SampleGridEnvironment_DC

Ensure the Server field had changed to the SampleGridEnvironment\_DC.

- \_\_\_ 7. Proceed to “Step 3: Summary”.
- \_\_\_ 8. Click "Finish" and wait for the installation to complete.

\_\_\_ 9. Save your changes. Be sure to check "Synchronize changes with Nodes"

Start the application

Restart the SampleGridEnvironment\_DC servers

\_\_\_ 10. Ensure all the SampleGridEnvironment... servers are started.

\_\_\_ a. In the administrative console Navigation panel, expand **Servers** and click **Application Servers**.

\_\_\_ b. Select any instances of SampleGridEnvironment\_DC\_hostXNodeNN that are not running.

Select	Name	Node	Version	Cluster Name	Status
<input type="checkbox"/>	LongRunningScheduler_DC_hostANode01	hostANode01	ND 6.1.0.3 XD 6.1.0.0	LongRunningScheduler_DC	➔
<input checked="" type="checkbox"/>	SampleGridEnvironment_DC_hostBNode01	hostBNode01	ND 6.1.0.3 XD 6.1.0.0	SampleGridEnvironment_DC	✘
<input checked="" type="checkbox"/>	SampleGridEnvironment_DC_hostCNode01	hostCNode01	ND 6.1.0.3 XD 6.1.0.0	SampleGridEnvironment_DC	✘
Total 3					

\_\_\_ c. Click **Start**.

\_\_\_ d. Wait for confirmation that the servers are started.

Run the application.

\_\_\_ 11. Edit the xJCL

To test the SimpleCI application, you will use the xJCL in <WAS\_HOME>/longRunning/SimpleCIxJCL.xml. Before submitting the job, edit the file and supply an output file name that is valid on the dynamic cluster where you have deployed the SimpleCI.ear application.

\_\_\_ a. Open Windows Explorer on hostA and navigate to C:\WebSphere\AppServer\longRunning.

\_\_\_ b. Right-click on SimpleCIxJCL.xml and click on Open With → WordPad (notepad does not format correctly)

\_\_\_ c. Find the line:

```
<prop name="outputFileName" value="<SimpleCI_output_file" />
```

and change it to (for example):

```
<prop name="outputFileName" value="C:/LabFilesXD/ComputeGridLab/SimpleCI.output.file" />
```

\_\_\_ d. Save the File SimpleCIxJCL.xml.

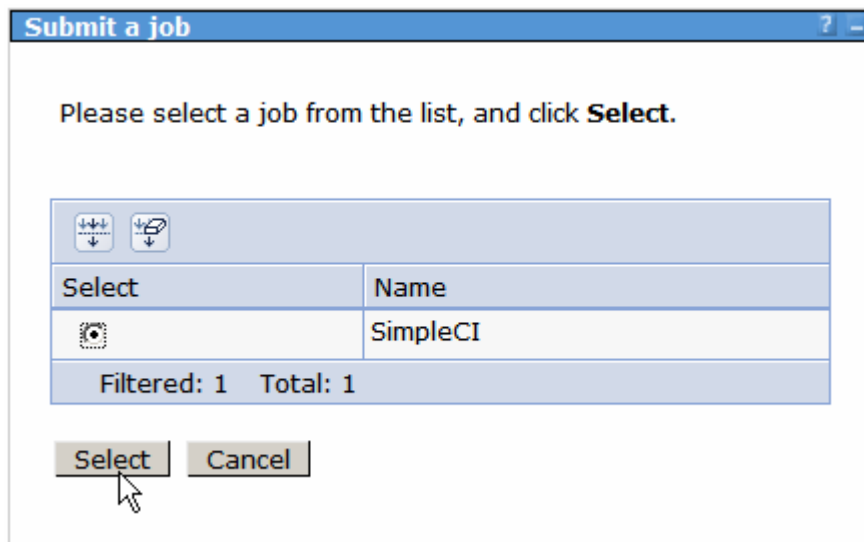
\_\_\_ 12. Save the SimpleCI job to the job repository

\_\_\_ a. Open internet Explorer and go to the Job Management Console. The URL should be http://<hosta>:< WC\_defaulthost >/jmc. For example http://hosta:9080/jmc

- \_\_\_ b. In the left panel expand "Job Repository" and click on "Save a job"
- \_\_\_ c. For "Job Name" enter "SimpleCI"
- \_\_\_ d. Click on **Browse**
- \_\_\_ e. Navigate to C:\WebSphere\AppServer\longRunning and select **SimpleClxJCL.xml**
- \_\_\_ f. Click on **Save**
- \_\_\_ g. Click on "View saved jobs" in the left panel.
- \_\_\_ h. Click on the job you just saved, SimpleCI, and inspect the xJCL.

\_\_\_ 13. Submit the saved SimpleCI job.

- \_\_\_ a. In the left panel click on "Submit a job" under "Job Management"
- \_\_\_ b. Check the radio button next to "Job repository" and then click on "Browse"
- \_\_\_ c. Check the box next to SimpleCI



- \_\_\_ d. Click "Select" and then "Submit".



Submit a job ? -

**Submit a job**  
Specify the job definition to submit as a job. The job definition can originate from the local file system or from the job repository. If a job has substitution properties without values, you will be prompted to specify them.

Local file system

\* Specify path to xJCL

Job repository

\* Specify job name


Update substitution properties


Delay submission

\* Start date (yyyy-MM-dd):  
 -  -

\* Start time (HH:mm:ss):  
 :  :

\_\_ e. In the left panel click on “View jobs” under “Job Management”

\_\_ f. Periodically click on the recycle icon **State**  until the job SimpleCI finishes.

Select	Job ID	Submitter	Last Update	State 	Node	Application Server
<input type="checkbox"/>	<a href="#">SimpleCIEar:0</a>		Thu Apr 12 15:02:42 CDT 2007	Ended	wsbeta157Node	SampleBatchEnvironment_DC_wsbeta157Node

Page: 1 of 4    Jump to page:     Filtered: 4    Total: 4

View the output from the application in the file specified in Step \_\_\_\_ 11. \_\_ c above. For instance, **C:/LabFilesXD/ComputeGridLab/SimpleCI.output.file** This file is located on the node where the job ran.

## Install tryit.class

Tryit is a simple Java program and demonstrates running a non-J2EE application. This job has a class file and an xJCL file that are located in C:/LabFilesXD/ComputeGridLab.

### 1. Edit the xJCL

To test the Tryit application, you will use the xJCL in C:\LabFilesXD\ComputeGridLab\tryit.xml. Before submitting the job, edit the file and supply a PATH and CLASSPATH that is valid on any of your systems.

\_\_\_ a. Open Windows Explorer on hostA and navigate to C:\LabFilesXD\ComputeGridLab.

\_\_\_ b. Right-click on tryit.xml and click on Open With → WordPad

\_\_\_ c. Set values to match these:

```
<?xml version="1.0" encoding="UTF-8" ?>
<job name="GridUtility-Test" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <job-step name="Step1">
    <env-entries>
      <env-var name="PATH" value="C:\windows;C:\WebSphere\AppServer\java\bin"/>
      <env-var name="CLASSPATH" value="C:\LabFilesXD\ComputeGridLab"/>
    </env-entries>
    <exec executable="java" >
      <arg line="tryit"/>
    </exec>
  </job-step>
</job>
```

\_\_\_ d. Note the differences between the xJCL to describe a WebSphere application and this non-WebSphere application.

1) The environment variables will set the PATH and CLASSPATH variables to access windows and Java libraries and your tryit.class file.

2) The command to run the program is “Java tryit”.

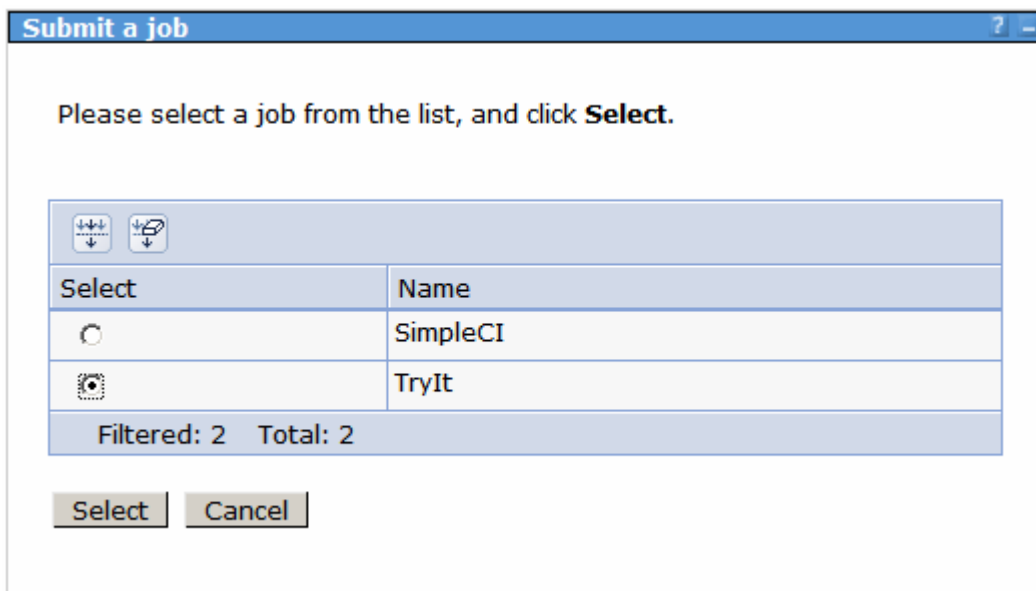
\_\_\_ e. Save tryit.xml.

### 2. Save the Tryit job to the job repository

\_\_\_ a. Open Internet Explorer and go to the Job Management console. The URL should be http://<hosta>:< WC\_defaulthost>/jmc.

\_\_\_ b. In the left panel click on “Save a job”

- \_\_\_ c. For “Job Name” enter “TryIt”
  - \_\_\_ d. Click on **Browse**
  - \_\_\_ e. Navigate to C:\LabFilesXD\ComputeGridLab and select **tryit.xml**
  - \_\_\_ f. Click on save
- \_\_\_ 3. Submit the saved TryIt job.
- \_\_\_ a. In the left panel click on “Submit a job” under “Job Management”
  - \_\_\_ b. Check the radio button next to “Job repository” and then click on “Browse”
  - \_\_\_ c. Check the box next to “TryIt”



- \_\_\_ d. Click “Select” and then “Submit”.

Submit a job
?

**Submit a job**  
Specify the job definition to submit as a job. The job definition can originate from the local file system or from the job repository. If a job has substitution properties without values, you will be prompted to specify them.

Local file system

\* Specify path to xJCL

Job repository

\* Specify job name

Update substitution properties

Delay submission

\* Start date (yyyy-MM-dd):

-  -

\* Start time (HH:mm:ss):

:  :

\_\_ e. In the left panel click on "View jobs" under "Job Management"

Since no special capabilities were required in the xJCL, the scheduler will assume the job can run on any of the available nodes in the cell.

\_\_ f. Periodically click on the recycle icon State  until the job "TryIt" finishes.

Select	Job ID	Submitter	Last Update	State	Node	Application Server
<input type="checkbox"/>	<a href="#">GridUtility-Test:1</a>		Mon Apr 23 13:52:54 CDT 2007	Ended	hostBNode01	nodeagent
Filtered: 1 Total: 1						

\_\_ g. Click on the Job ID for the job that just completed. It should contain something like:

```

CWXDG5005I: [04/23/07 13:52:34:015 CDT] Setting up Grid Utility job GridUtility-Test:1 for
execution: [jobClass Default] [jobName GridUtility-Test] [user UNAUTHENTICATED] [logDirectory 23032007_135209]
CWXDG5008I: [04/23/07 13:52:34:124 CDT] Beginning Grid Utility Job GridUtility-Test:1 execution.
CWXDG5009I: [04/23/07 13:52:34:124 CDT] Beginning step Step1 execution.
System.err: [04/23/07 13:52:54:515 CDT] JAVA FOO
System.err: [04/23/07 13:52:54:515 CDT] s is null
System.err: [04/23/07 13:52:54:515 CDT] s is not null
System.err: [04/23/07 13:52:54:515 CDT] waiting ...
System.err: [04/23/07 13:52:54:515 CDT] Wait over.
System.err: [04/23/07 13:52:54:515 CDT] Goodbye
CWXDG5011I: [04/23/07 13:52:54:530 CDT] Grid Utility Job step Step1 ended.
CWXDG5013I: [04/23/07 13:52:54:546 CDT] Grid Utility Job GridUtility-Test:1 ended.

```

- \_\_\_ h. Note that you can also view the job log directly. Open Windows Explorer on the node where tryit ran and “cd C:\<WAS\_HOME>\profiles\<node name>\joblogs\<grid test name>” where <node name> is the node running the scheduler, and <grid test name> is the Job ID from the job management console.

## Install MandelbrotCI.ear

MandelbrotCI is a compute-intensive application with a Web interface that computes and renders Mandelbrot fractal images. The application contains two modules: a Web module that provides the user interface and an EJB module that contains the compute-intensive logic. You will deploy the Web module to the same dynamic cluster as the job scheduler and the EJB module to the execution environment dynamic cluster.

### Verify virtual hosts

In this lab you will access the Mandelbrot servlet component directly from a browser. Since you will deploy this component to the same dynamic cluster as the scheduler, you need to take note again of the scheduler’s WC\_defaulthost port address and the node on which it is running. The scheduler’s installation should have created an alias for the scheduler server’s WC\_defaulthost port. You recorded this port in the table on page 11. That node and port address will be needed when invoking the Madelbrot servlet from the browser.

- \_\_\_ 1. Find the Application Server’s port
- \_\_\_ a. In the administrative console navigation tree, expand "Servers" and click on "Application Servers".
  - \_\_\_ b. Select LongRunningScheduler\_DC\_hostANode01.
  - \_\_\_ c. Under Communications”, expand “Ports”.

Communications	
Ports	
Port Name	Port
BOOTSTRAP_ADDRESS	9811
SOAP_CONNECTOR_ADDRESS	8884
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	9404
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	9405
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	9406
WC_adminhost	9061
WC_defaulthost	9080
DCS_UNICAST_ADDRESS	9357
WC_adminhost_secure	9044
WC_defaulthost_secure	9444
SIB_ENDPOINT_ADDRESS	7276
SIB_ENDPOINT_SECURE_ADDRESS	7286
SIB_MQ_ENDPOINT_ADDRESS	5558
SIB_MQ_ENDPOINT_SECURE_ADDRESS	5578
ORB_LISTENER_ADDRESS	0

\_\_\_ d. Note the Port associated with WC\_defaulthost. In this example it is 9080.

Server	Port
LongRunningScheduler_DC_hostANode01	

Install the Application

- \_\_\_ 2. In the administrative console navigation tree, expand "Applications" and click on "Install New Application".
- \_\_\_ 3. Select "Remote file system", and Browse to hostACellManager01 → "C:\WebSphere\AppServer\installableApps and select "MandelbrotCI.ear" and click "OK".
- \_\_\_ 4. Click "Next".

If you receive an Application Security warning, click "Continue".

- \_\_\_ 5. In “Step 1: Select installation Options”, clear the check box for “Deploy enterprise beans”

**Select installation options**

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

Pre-compile JSP

Directory to install application

Distribute application

Use Binary Configuration

Deploy enterprise beans

Application name

Application Edition

Edition Description

Create MBeans for resources

Enable class reloading

Reload interval in seconds

Deploy Web services

Validate Input off/warn/fail

Process embedded configuration

- \_\_\_ 6. Press “Next”

The MandelbrotCI application contains two modules: a Web module that provides the user interface and an EJB module that contains the compute-intensive logic. You will deploy the Web module to the same dynamic cluster as the job scheduler and the EJB module to the execution environment dynamic cluster.

- \_\_\_ 7. In “Step 2: Map modules to servers”, select MandelbrotCIEJB and SampleGridEnvironment\_DC. Press “Apply”

Clusters and Servers:

WebSphere:cell=hostACell01,cluster=SampleGridEnvironment\_DC  
 WebSphere:cell=hostACell01,cluster=LongRunningScheduler\_DC

Apply

Select	Module	URI	Server
<input checked="" type="checkbox"/>	MandelbrotCIEJB	MandelbrotCIEJB.jar,META-INF/ejb-jar.xml	WebSphere:cell=hostACell01,cluster=SampleGridEnvironment_DC
<input type="checkbox"/>	MandelbrotCIWeb	MandelbrotCIWeb.war,WEB-INF/web.xml	WebSphere:cell=hostACell01,cluster=LongRunningScheduler_DC

- \_\_\_ 8. Select MandelbrotCIWEB and LongRunningScheduler\_DC. Press “Apply”

Clusters and Servers:

WebSphere:cell=hostACell01,cluster=SampleGridEnvironment\_DC  
 WebSphere:cell=hostACell01,cluster=LongRunningScheduler\_DC

Apply

Select	Module	URI	Server
<input type="checkbox"/>	MandelbrotCIEJB	MandelbrotCIEJB.jar,META-INF/ejb-jar.xml	WebSphere:cell=hostACell01,cluster=SampleGridEnvironment_DC
<input checked="" type="checkbox"/>	MandelbrotCIWeb	MandelbrotCIWeb.war,WEB-INF/web.xml	WebSphere:cell=hostACell01,cluster=LongRunningScheduler_DC

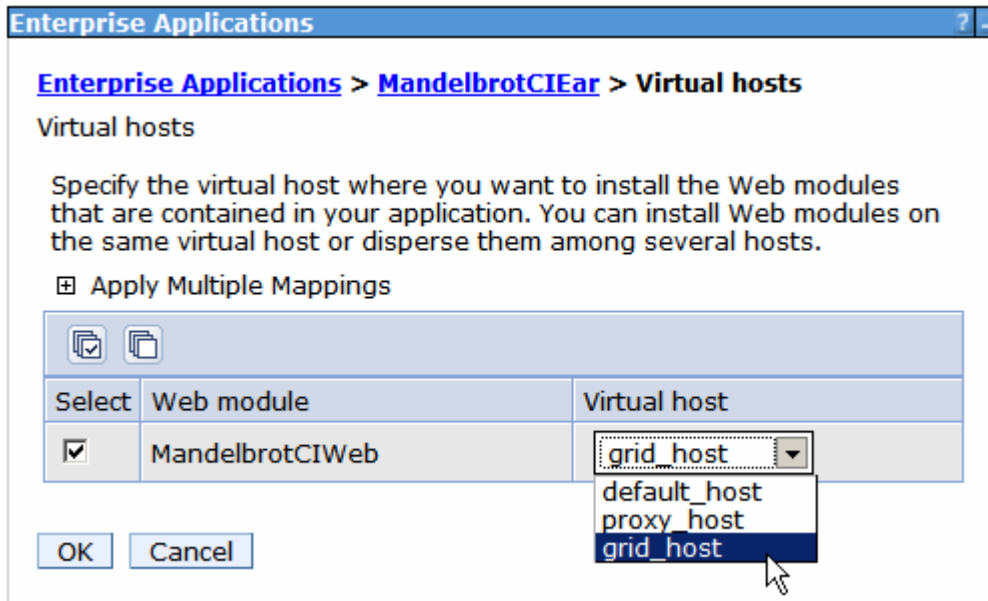
- \_\_\_ 9. Click on “Step 3: Summary”.
- \_\_\_ 10. Click "Finish" and wait for the installation to complete.

Since you deployed this application to the job scheduler server, you must also set the application’s virtual host to match the scheduler’s

- \_\_\_ 11. In the administrative console, expand Applications and click on Enterprise Applications.
- \_\_\_ 12. Click on MandelbrotCIEar to view the application’s details.
- \_\_\_ 13. Under Web Module Properties, click Virtual Hosts




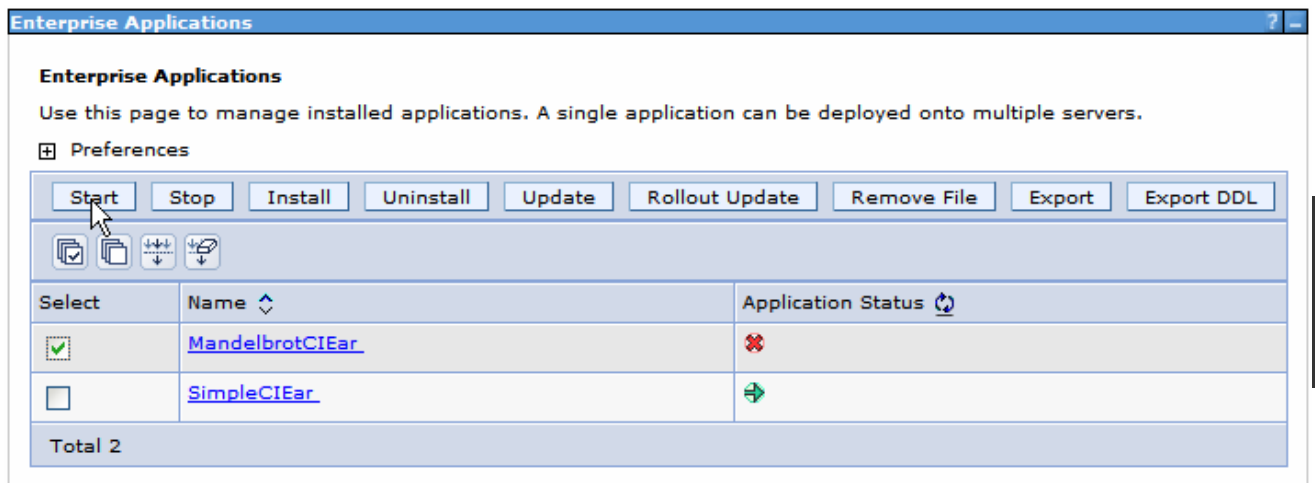
- \_\_\_ 14. Click the box next to MandelbrotCIWeb and select grid\_host from the list



- \_\_\_ 15. Click OK
- \_\_\_ 16. Save your changes. Be sure to check "Synchronize changes with Nodes"

Start the application

- \_\_\_ 17. In the administrative console navigation tree, expand "Applications" and click on "Enterprise Application"
- \_\_\_ 18. Check the boxes next to MandelbrotCIEar (if it's status is not started ) and press "Start"



- \_\_\_ 19. Wait for the application to start.

## Run the application

Once installed, the URL for the Web GUI is `http://<host>:<port>/mandelbrotci/gui`.

Note: to invoke the Mandelbrot servlet, use the **Long Running Scheduler's** Application Server's host and WC\_defaulthost port, which discussed on page 29.

- \_\_\_ 20. Start the Mandelbrot application.
- \_\_\_ 21. Open a browser to URL <http://<hosta>:<port>/mandelbrotci/gui>.  
For instance, `http://hostA:9080/mandelbrot/gui`
- \_\_\_ 22. Click on "(Re)start computation"

---

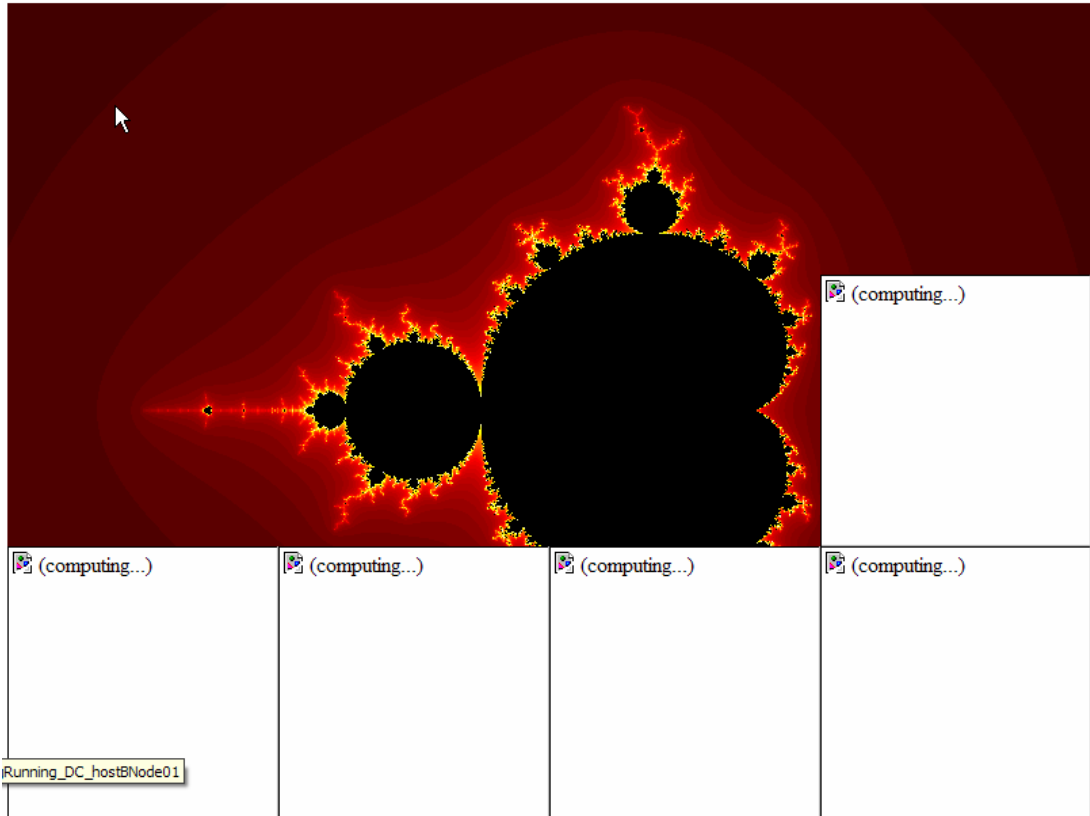
Be patient. The grid might not fill up uniformly. By default there are 10 initiators, so most likely 10 squares will fill up first. After a while, the remaining 2 will get scheduled and the grid will complete. This behavior can be managed by adding a "Custom Property" under SampleGridEnvironment\_DC of "com.ibm.websphere.longrun.EndPointCapacity" with a value greater than 12, for this case.

The procedure to set this property is:

- \_\_\_ a. From the administrative console, open Servers→Dynamic Clusters→SampleGridEnvironment\_DC
  - \_\_\_ b. Under Additional Properties, click Custom Properties
  - \_\_\_ c. Add this line: "com.ibm.websphere.longrun.EndPointCapacity" == "20"
  - \_\_\_ d. Restart the long running execution environment servers.
-

Lab exercise: Compute grid

The application will perform its calculations in “chunks,” filling in each section of the picture as the calculation job completes:



23. The calculations are performed as a series of compute intensive jobs. You can see this in the administrative console, by navigating to the Job Management Console (<http://hostA:9080/jmc>). Expand Job Management and click View Jobs. Notice that a job has been added for each square in the application (see Job Management Console).

Select	Job ID	Submitter	Last Update	State	Node	App Server
<input type="checkbox"/>	<a href="#">MandelbrotCIEar:10</a>		Thu Oct 19 20:14:43 CDT 2006	Ended	hostBNode01	SampleBatchEnvironment_DC_hostBNode01
<input type="checkbox"/>	<a href="#">MandelbrotCIEar:11</a>		Thu Oct 19 20:14:43 CDT 2006	Ended	hostBNode01	SampleBatchEnvironment_DC_hostBNode01
<input type="checkbox"/>	<a href="#">MandelbrotCIEar:12</a>		Thu Oct 19 20:14:43 CDT 2006	Ended	hostBNode01	SampleBatchEnvironment_DC_hostBNode01
<input type="checkbox"/>	<a href="#">MandelbrotCIEar:6</a>		Thu Oct 19 20:14:42 CDT 2006	Ended	hostCNode01	SampleBatchEnvironment_DC_hostCNode01
<input type="checkbox"/>	<a href="#">MandelbrotCIEar:7</a>		Thu Oct 19 20:14:43 CDT 2006	Ended	hostBNode01	SampleBatchEnvironment_DC_hostBNode01
<input type="checkbox"/>	<a href="#">MandelbrotCIEar:8</a>		Thu Oct 19 20:14:42 CDT 2006	Ended	hostCNode01	SampleBatchEnvironment_DC_hostCNode01
<input type="checkbox"/>	<a href="#">MandelbrotCIEar:9</a>		Thu Oct 19 20:14:42 CDT 2006	Ended	hostBNode01	SampleBatchEnvironment_DC_hostBNode01

## Install PostingsSample.ear

PostingsSample is a transactional batch application that mimics account transaction processing.

### Create the sample database

PostingsSample.ear contains entity beans. The DDL files to create the required database tables and tablespaces are in <WAS\_install\_root>/longRunning/CreatePostingsTablesXxx.ddl and <WAS\_install\_root>/longRunning/CreatePostingsTablespaceXxx.ddl, where Xxx denotes the type of database manager for which the DDL file is intended. You will need to create the database tables and define a data source for the database before installing the PostingsSample application.

For this lab you will use the Derby Network database engine included with WebSphere. Some of the parameters may be different if you use a different database. Create the database in C:\LabFilesXD\ComputeGridLab\databases on hostB.

1. Create the POSTINGS database. At a Windows command prompt on hostB, change the current directory to C:\LabFilesXD\ComputeGridLab\databases and issue this command:

```
createPostingsDB.bat
```

After creating the database, you need to define the database to WebSphere Extended Deployment. You will define the database at the dynamic cluster level where the application will be installed.

### Configure Derby

In order to enable the batch environment on multiple servers in a cluster, you need to configure a Derby Network Server: This framework was a new feature in Derby Version 10.1.2.2, and removes many limitations that existed in earlier versions of Derby:

These steps describe how to configure and run the Network Server framework:

- \_\_\_ 2. On hostB, and hostC, edit C:\WebSphere\AppServer\derby\derby.properties. Uncomment the line to allow connections to the derby network server.

```
derby.drda.host=0.0.0.0
```

- \_\_\_ 3. Start the Derby Network server on hostB.

At a Windows command prompt on hostB, change to directory C:\WebSphere\AppServer\derby\bin\networkServer and run startNetworkServer.bat.

### Create a data source

- \_\_\_ 4. Define a Derby Network server JDBC provider using Universal JDBC driver to connect Derby with WebSphere Application Server using the Network Server framework.
  - \_\_\_ a. From the administrative console, expand “Resources” in the navigation tree, then expand JDBC, and click on “JDBC Providers”.
  - \_\_\_ b. From the **scope** pull down select cluster Cluster=SampleGridEnvironment\_DC.
- \_\_\_ 5. Click on “New”.

\_\_ a. Select these values:

**database type**

Derby

**Provider type**

Derby Network Server Using Derby Client

**Implementation type**

XA data source

**Name**

Sample LongRunning\_DS Derby Network Server (XA)

**Step 1: Create new JDBC provider**

Step 2: Enter database class path information

Step 3: Summary

**Create new JDBC provider**

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope  
cells:hostACell01:clusters:SampleGridEnvironment\_DC

\* Database type  
Derby

\* Provider type  
Derby Network Server Using Derby Client

\* Implementation type  
XA data source

\* Name  
Sample LongRunning\_DS Derby Network Server (XA)

Description  
Derby Network Server (XA) Provider that uses the Derby Client. This provider is only configurable in version 6.1 and later nodes

Next Cancel

\_\_ b. Click on "Next" then "Finish"

\_\_ c. Save your changes. Be sure to check "Synchronize changes with nodes"

- \_\_\_ 6. Select the provider you just created
- \_\_\_ 7. Under "Additional Properties", click on "Data sources".



- \_\_\_ 8. Click on "New".
- \_\_\_ 9. Enter these values:

**Data source name**

**POSTINGS**, or another name of your choosing. Note that if you are defining per-node datasources, each must have a unique name.

**JNDI name**

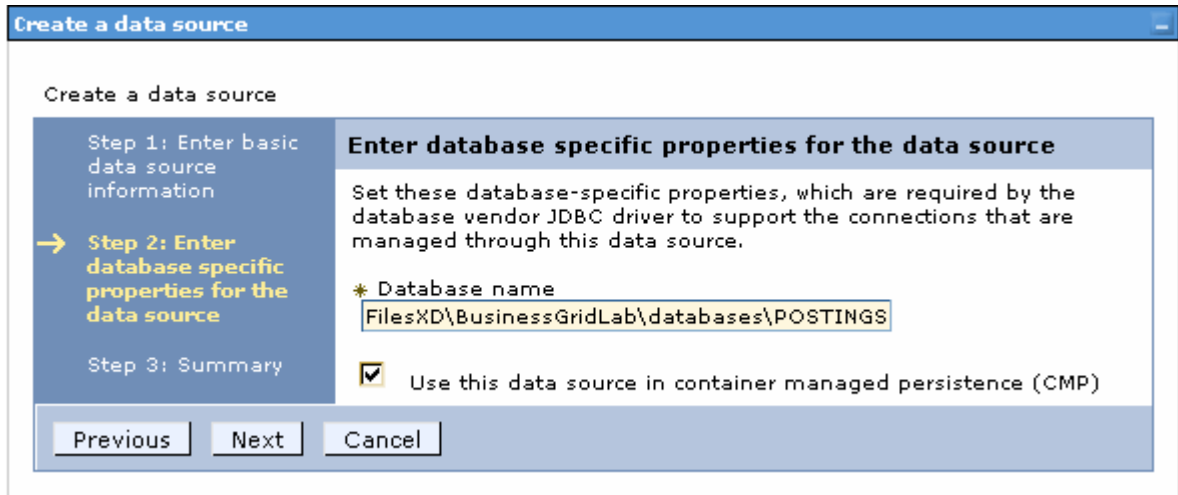
**jdbc/postings**, or another JNDI name of your choosing for the data source.

- \_\_\_ 10. Click "Next".

\_\_\_ 11. Enter

**Database name**

**C:\LabFilesXD\ComputeGridLab\databases\POSTINGS**



\_\_\_ 12. Press “Next” and “Finish”

\_\_\_ 13. Click on the data source you just created, POSTINGS

\_\_\_ 14. Under Additional Properties, click on Custom properties

\_\_\_ a. Click serverName

1) Change the Value to **hostB**

2) Click OK

\_\_\_ b. Ensure the portNumber is 1527

\_\_\_ 15. Save your changes. Be sure to check “Synchronize changes with Nodes”

\_\_\_ 16. Test the database connection

\_\_\_ a. From the administrative console, Expand “Resources”, then expand “JDBC” click on “Data Sources”

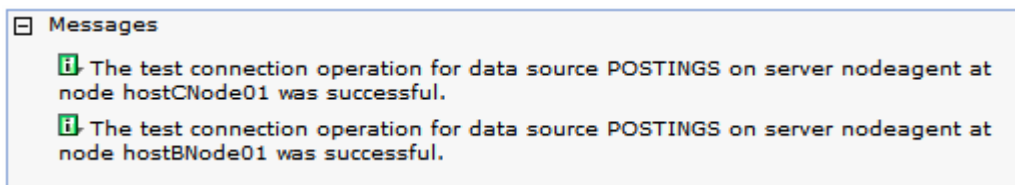
\_\_\_ b. Click on the JDBC provider “SampleGridEnvironment\_DS Derby Network Server”.

\_\_\_ c. Check the box next to POSTINGS and press “Test connection”



Select	Name	JNDI name	Scope	Provider
<input type="checkbox"/>	<a href="#">LREE</a>	jdbc/lree	Cell=hostACell01	Default Grid Derby JDBC Provider
<input type="checkbox"/>	<a href="#">LRSCHED</a>	jdbc/lrsched	Cell=hostACell01	Default Grid Derby JDBC Provider
<input checked="" type="checkbox"/>	<a href="#">POSTINGS</a>	jdbc/postings	Cluster=SampleGridEnvironment_DC	Smaple LongRunning_DS Derby Network Server (XA)

\_\_\_ d. If you configured the JDBC provider and data source correctly, you should see messages similar to:



Install the application

- \_\_\_ 17. In the administrative console navigation tree, expand "Applications" and click on "Install New Application".
- \_\_\_ 18. Select "Remote file system", and Browse to hostACellManager01 → "C:\WebSphere\AppServer\installableApps" and select "PostingsSample.ear" and click "OK".

- \_\_\_ 19. Select the check box for "Show me all installation options and parameters."

**Preparing for the application installation** ? -

Specify the EAR, WAR, JAR, or SAR module to upload and install.

**Path to the new application**

Local file system

Full path

Remote file system

Full path

Context root  Used only for standalone Web modules (.war files) and SIP modules (.sar files)

**How do you want to install the application?**

Prompt me only when additional information is required.

Show me all installation options and parameters.

- \_\_\_ 20. Click "Next" twice.

If you receive an Application Security Warning, click "Continue".

- \_\_\_ 21. Accept defaults in "Step 1: Select installation Options"

- \_\_\_ 22. Click "Next"

- \_\_\_ 23. In "Step 2: Map modules to servers", select module PostingsJobEJBs and dynamic cluster SampleGridEnvironment\_DC. Click "Apply"

**Map modules to servers**

Specify targets such as application servers or clusters of application servers where you want to install your application. Modules can be installed on the same application server or dispersed among several Web servers as targets that serve as routers for requests to this application. The plug-in configuration Web server is generated, based on the applications that are routed through.

Clusters and Servers:

- WebSphere:cell=hostACell01,cluster=SampleGridEnvironment\_DC
- WebSphere:cell=hostACell01,cluster=LongRunningScheduler\_DC
- WebSphere:cell=hostACell01,node=wsbeta177,server=middlewareagent
- WebSphere:cell=hostACell01,node=wsbeta176,server=middlewareagent

Apply

Select	Module	URI	Server
<input checked="" type="checkbox"/>	PostingsJobEJBs	PostingsSampleEJBs.jar,META-INF/ejb-jar.xml	WebSphere:cell=hostACe

- \_\_\_ 24. In " Step 3: Provide options to perform the EJB Deploy":
- \_\_\_ a. Select the type of database you used for your execution environment database in the "Deploy EJB option - Database type" field. For Derby, choose "DERBY\_V10".
  - \_\_\_ b. Enter "POSTINGSSHEMA" for "Deploy EJB option - Database schema".

**Provide options to perform the EJB Deploy**

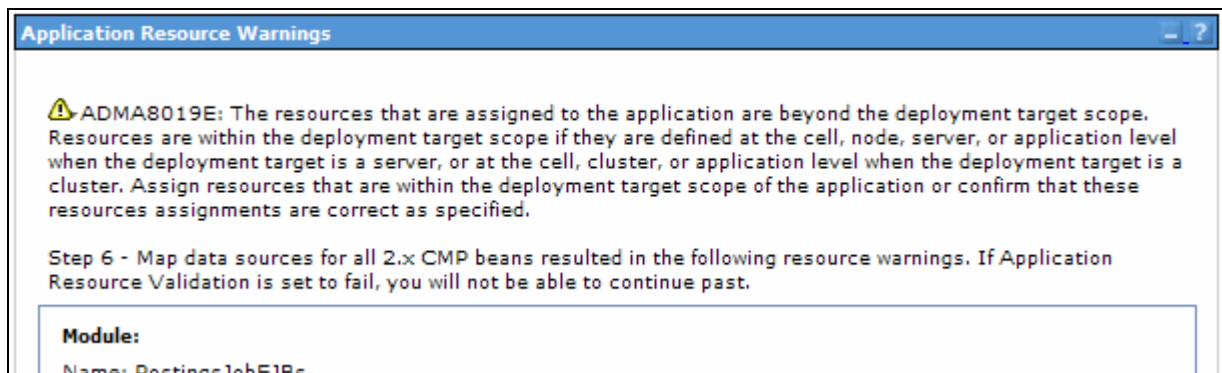
Specify the options to deploy enterprise beans. Select database type only when all of the modules are mapped to the same database type. If some modules map to a different backend ID, set the database type blank so that the Select current backend ID panel is displayed.

EJB Deployment Options	Enable
Deploy EJB option - Class path	<input type="text"/>
Deploy EJB option - RMIC	<input type="text"/>
Deploy EJB option - Database type	DERBY_V10
Deploy EJB option - Database schema	POSTINGSSHEMA

- \_\_\_ c. Click on "Step 8: Map data sources for all 2.x CMP beans":
  - 1) Ensure the target Resource JNDI Name for each EJB is jdbc/postings.

Select	EJB	EJB module	URI	Target Resource JNDI Name	Resource authorization
<input checked="" type="checkbox"/>	Account	PostingsJobEJBs	PostingsSampleEJBs.jar,META-INF/ejb-jar.xml	jdbc/postings Browse...	Resource authorization: Per application
<input checked="" type="checkbox"/>	DataCreationBean	PostingsJobEJBs	PostingsSampleEJBs.jar,META-INF/ejb-jar.xml	jdbc/postings Browse...	Resource authorization: Per application
<input checked="" type="checkbox"/>	PostingAccountData	PostingsJobEJBs	PostingsSampleEJBs.jar,META-INF/ejb-jar.xml	jdbc/postings Browse...	Resource authorization: Per application
<input checked="" type="checkbox"/>	OverdraftAccountPosting	PostingsJobEJBs	PostingsSampleEJBs.jar,META-INF/ejb-jar.xml	jdbc/postings Browse...	Resource authorization: Per application

\_\_ d. Click on the final step "Summary". If you receive ADMA8019E error, ignore it and click "Continue"

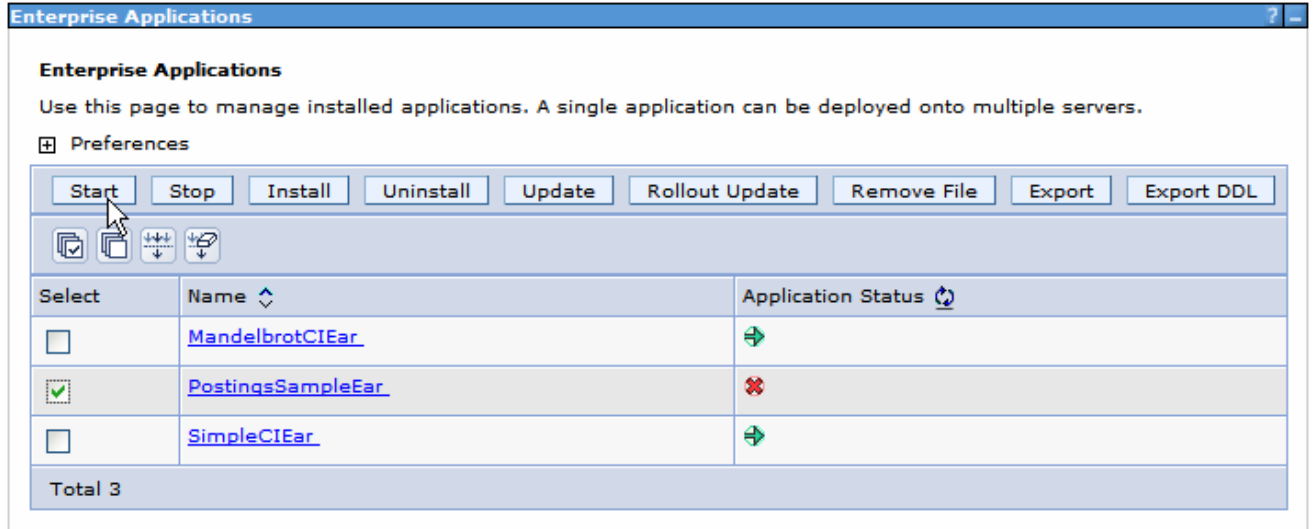


\_\_\_ 25. Click "Finish" and wait for the installation to complete.

\_\_\_ 26. Save your changes. Be sure to check "Synchronize changes with nodes"

Start the application

- \_\_\_ 27. In the administrative console navigation tree, expand "Applications" and click on "Enterprise Application"
- \_\_\_ 28. Check the boxes next to PostingsSampleEar and press "Start"



- \_\_\_ 29. Wait for the applications to start

Run the application.


To test the PostingsSample application, you will use the xJCL in <WAS\_HOME>/longRunning/postingSampleXJCL.xml.

- \_\_\_ 30. Submit the job without saving the xJCL in the job repository
  - \_\_\_ a. Open internet Explorer and go to the Job Management Console. The URL should be http://<hosta>:< WC\_defaulthost >/jmc.

- \_\_\_ b. In the left panel expand Job Management and click on “Submit a new job”
- \_\_\_ c. Click Local file system
- \_\_\_ d. Browse to C:\WebSphere\AppServer\longRunning\postingSampleXJCL.xml

- \_\_\_ e. Click Submit

\_\_\_ 31. In the left panel click on “View jobs” under “Job Management”

- \_\_\_ a. Periodically click on the recycle icon **State**  until the job “PostingsSampleEar” finishes.
- \_\_\_ b. Determine which server the job ran on It is listed in the Job Management Console:

Select	Job ID	Submitter	Last Update	State	Node	App Server
<input type="checkbox"/>	<a href="#">PostingsSampleEar:16</a>		Sun Oct 22 22:21:00 CDT 2006	Submitted	hostBNode01	SampleGridEnvironment_DC_hostBNode01
<input type="checkbox"/>	<a href="#">SimpleCEar:0</a>		Sun Oct 22 17:54:42 CDT 2006	Ended	hostCNode01	SampleGridEnvironment_DC_hostCNode01

Filtered: 2 Total: 2

- \_\_\_ c. Open a windows explorer on that host and “cd \WebSphere\AppServer\temp\postings”. View the file with notepad. The contents should look similar to this:

```
0,4382,387.0
0,5846,328.0
0,5086,446.0
0,8366,288.0
0,8868,166.0
```

## What you did in this exercise

In this exercise you configured a long-running scheduler, installed the batch run-time in a dynamic cluster, and installed and ran the sample grid utility applications.

This page is left intentionally blank.