IBM Software
WebSphere

WebSphere Lab Jam

Application Infrastructure

WebSphere eXtreme Scale

Lab Exercise

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

Lab 1 Getting Started with eXtreme Scale Deployment and Development

IBM WebSphere eXtreme Scale 6.1 is a highly scalable and reliable elastic data grid designed to provide nearly limitless potential capacity and performance while focusing on simplicity of deployment, completeness of features and seamless integration with processing environments of all shapes and sizes. In this lab we will walk through the very first steps to understanding, deploying and working with a simple eXtreme Scale data grid. We will utilize the freely available trial eXtreme Scale trial download as well as an eclipse development environment to accomplish some key learning tasks and first steps.

- Deploy your first grid
- Add and remove objects from the grid using the provided client application
- Create an Eclipse development environment to start writing your own grid applications
- Introduction to the xsadmin tool provided to monitor the characteristics of a deployed grid.

1.1 Installing the WebSphere eXtreme Scale trial download

During this lab you will work with WebSphere eXtreme Scale in a *stand-alone* environment. A stand-alone environment is one that does not contain an existing WebSphere Application Server or other Java™ Platform, Enterprise Edition installation. The stand-alone environment requires only two products:

- Java Runtime Environment
- WebSphere eXtreme Scale

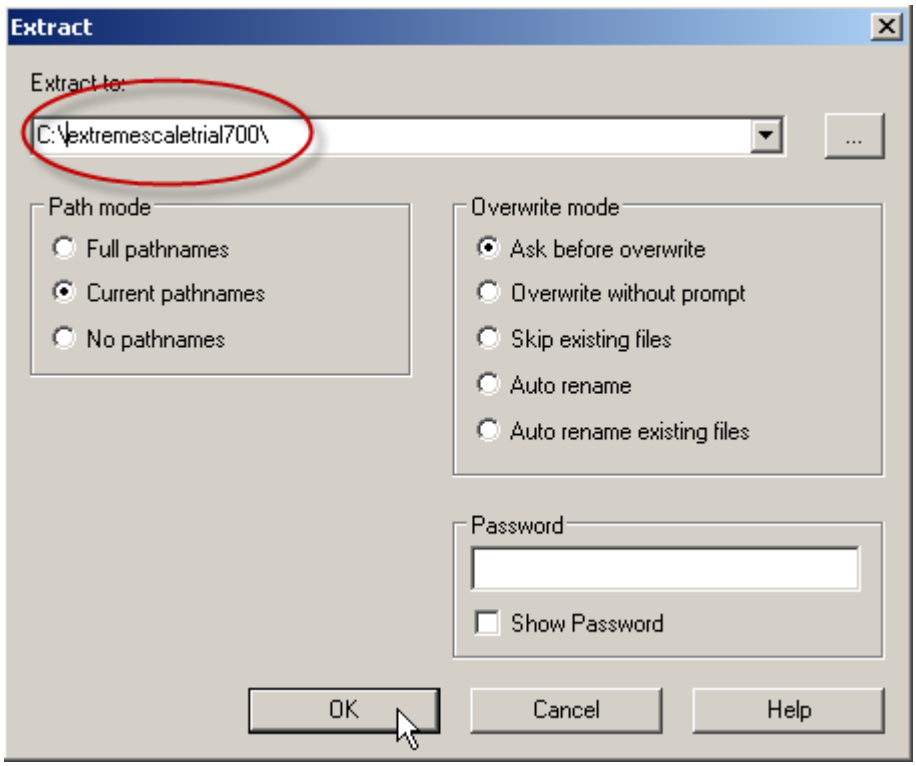
A popular method of obtaining WebSphere eXtreme Scale is downloading the free trial version. The trial download is publically available at:
http://www.ibm.com/developerworks/downloads/ws/wsdg/learn.html

The trial download enables the WebSphere eXtreme Scale features with a lightweight footprint. It is limited to running WebSphere eXtreme Scale servers for eight hours at a time, before requiring a restart. Installing the trial download is a simple matter of unzipping it to the file system. After this quick setup, you are ready to run and use the product. You will now perform these first steps and setup WebSphere eXtreme Scale.

__1. The trial version of WebSphere eXtreme Scale is named **extremescaletrial700.zip** and a copy is available on your Windows® desktop, as shown below.

__2. Right click on extremescaletrial700.zip and select **7-Zip->Extract files…**

__3. Specify **C:\extremescaletrial700** in the Extract to field. Click **OK**.



__4. The WebSphere eXtreme Scale v7.0 trail software is now ready to use.

1.2 Deploying your first grid

In this section you will quickly tailor the getting started example and launch a catalog server and two object grid container servers.

__1. Open a command line window. You can do this on the provided virtual image by clicking on the command prompt image on the Quick Launch tool bar at the bottom of the desktop as shown below:

__5. Change the current directory to the getting started of the trial download. Issue the following command:

cd C:\extremescaletrial700\ObjectGrid\gettingstarted

__2. Only one customization is needed for a new WebSphere eXtreme Scale stand-alone environment -- setting the `JAVA_HOME` environment variable to reference a valid JDK™ or JRE™ Version 5.0 or later installation directory. Issue the command:

notepad env.bat

__3. In the `env.bat` file, locate where the `JAVA_HOME` environment variable is set. Change it to refer to the JRE 6.0 installation directory `C:\java\jre6`. The line should be modified as follows:

```
REM ************************************************************
REM *** Change or set JAVA_HOME to match your environment ***
REM ************************************************************
if not defined JAVA_HOME SET JAVA_HOME=C:\Java\jre6
```

__6. **Save** changes to env.bat file and close the Notepad editor.

__7. The first step to deploying an eXtreme Scale data grid to is start a catalog service, which will act as coordinator for all other JVM's participating in the grid and manage configuration information. Start this service by issuing the command:

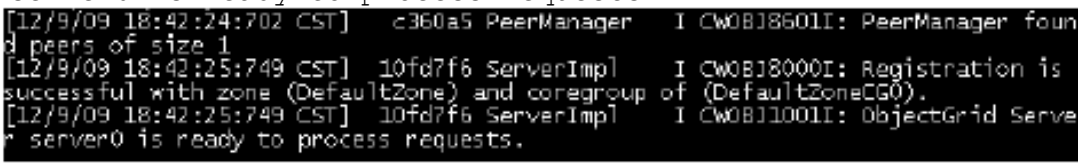
runcat.bat

The catalog service is running when you see the 'ObjectGrid Server cs0 is ready to process requests.' log message on the screen.

__4. A *container* is a server Java Virtual Machine (JVM™) that stores and caches application data for the grid. The application data is generally broken into parts, which are called *partitions*, and hosted across multiple containers. Each container hosts a subset of the complete application data. You can start as many containers as you wish, WebSphere eXtreme Scale automatically spreads the partitions out as more containers become available. As additional containers are started, they automatically register themselves with the catalog service allowing them to cooperate in providing grid services. This increases both grid capacity and reliability. To start the first container, return to your original Command Prompt window and issue:

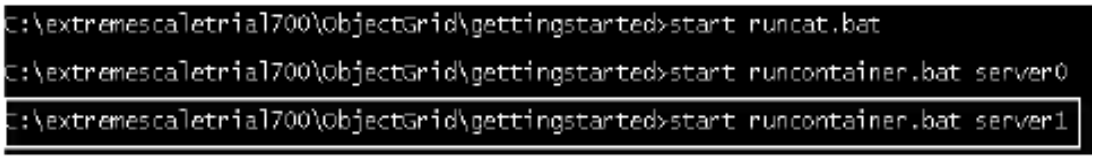
start runcontainer.bat server0

__8. Similar to the catalog service, the server is ready when you see `ObjectGrid Server server0 is ready to process requests`



__9. In order to demonstrate the scalability and reliability features of WebSphere eXtreme Scale, start a second container. Using your original Command Prompt window, issue the command:

start runcontainer.bat server1



1.3 WebSphere eXtreme Scale client

WebSphere eXtreme Scale clients connect to a catalog service, retrieve a description of the server topology, and then communicate directly to each container server as needed. When the server topology changes because new servers are added or existing servers have failed, the client is automatically routed to the appropriate server that is hosting the data.

__1. You will now test the grid using a client program supplied in the gettingstarted directory. Use the client program to insert, update and delete data from the newly deployed grid. To insert a key/value pair into the grid, return to your command prompt window and issue the command:

runclient.bat i key1 helloworld

You should see `SUCCESS: Inserted helloworld with key key1`

__10. Now use the client to retrieve that information from the grid. Issue the command:

runclient.bat g key1

__11. You can easily update the information in the grid with the update command:

runclient.bat u key1 goodbyeWorld

__12. Retrieve the update value:

runclient.bat g key1

__13. Finally we can delete the value. Issue the command:

runclient.bat d key1

__14. These are the basic functions of the sample client program and a simple demonstration of the concepts behind an eXtreme Scale data grid. Take some time to experiment by inserting your own key/value pairs and observing the results. For the sake of the sample avoid using spaces in your keys and values. While the product is perfectly capable of this and many other things, we wanted to keep the runclient scripts readable and learnable so avoided some standard error checking. Once you have loaded some data into your grid proceed to the next section.

__15. **Leave your catalog service and grid JVM's running!** We'll be using them to execute similar client operations from within Eclipse in the next section,

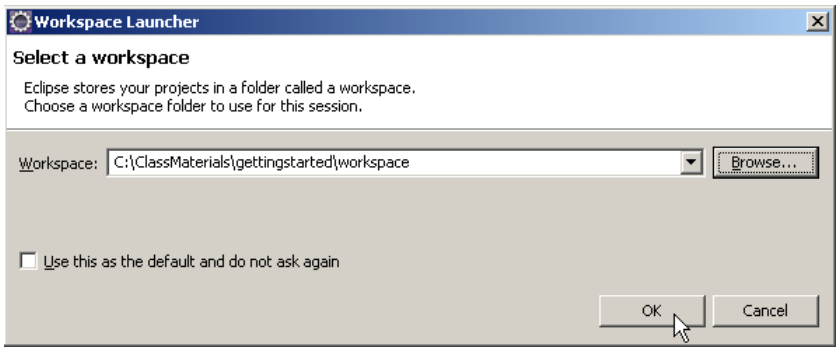
1.4 Configure an eclipse development environment

The demonstration above utilized simple scripts to invoke a basic java application which has been developed using the eXtreme Scale API's to perform simple grid tasks. In your own use of WebSphere eXtreme Scale, you will integrate these API's into your own java applications. In this section we will walk you through beginning this process by importing the proper eXtreme Scale libraries into an Eclipse project and using them to run the sample client application.

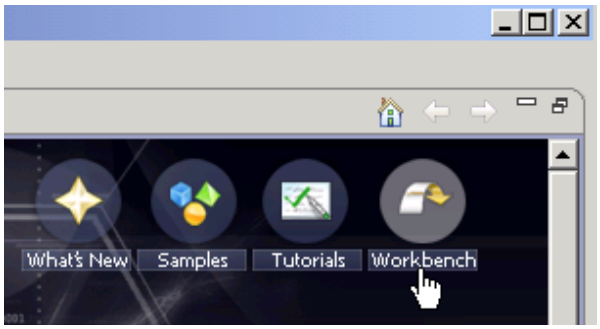
__1. Open eclipse with the icon provided on the desktop.

__2. You will be prompted for the location for a workspace. We are starting from scratch here so any location that doesn't exist will work fine. For the purposes of this exercise use the directory:

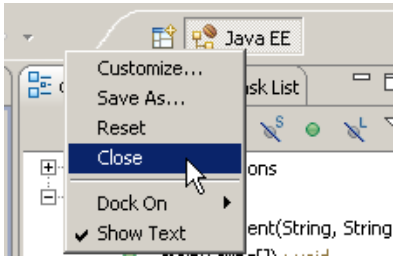
C:\ClassMaterials\gettingstarted\workspace



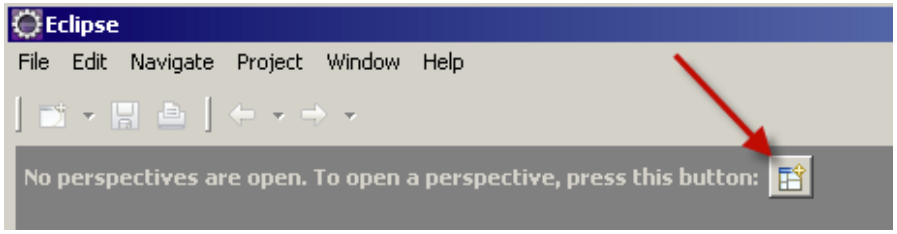
__16. From the upper-right of the workspace, click the **Workbench** icon.



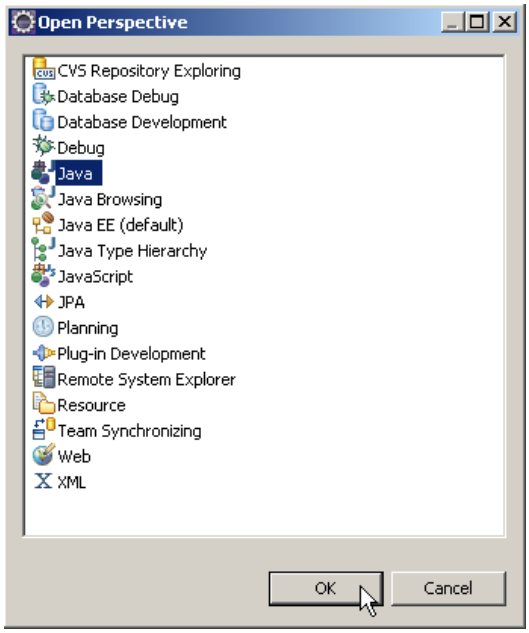
__17. Since you will be working with this project in a Java Standard Edition(JSE) environment, close the Java Enterprise Edition perspective. Right-click on the **Java EE** icon in the top right corner, and select **Close**.



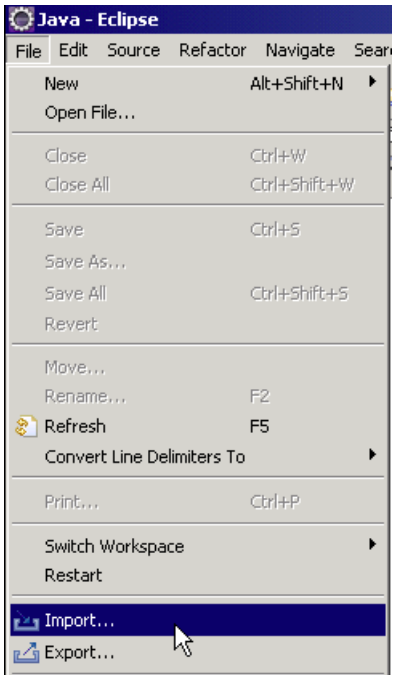
__3. Open a perspective by pressing the button in the middle of the workspace.



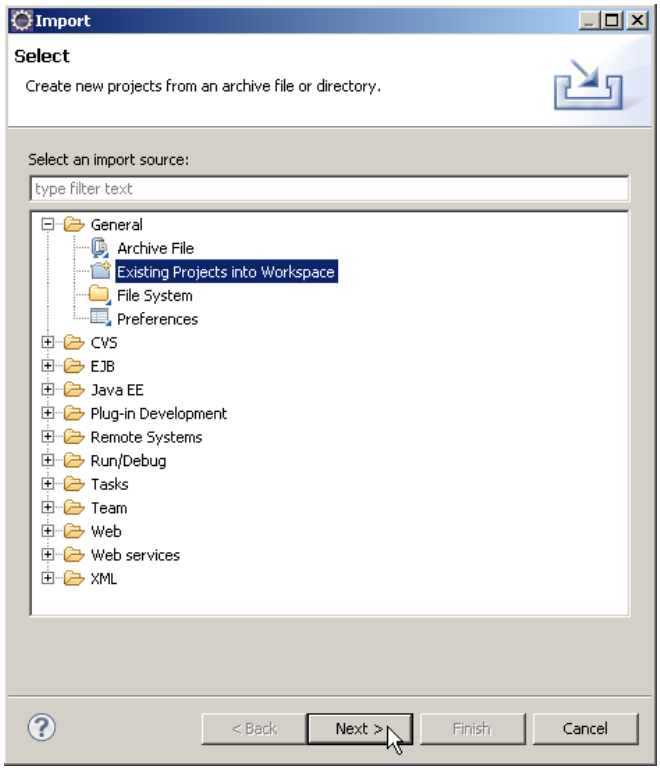
__18. Select **Java** and click **OK.**



__19. We are going to use eclipse project metadata that is provided with the eXtreme Scale trial package gettingstarted to create this project. Not only will it simplify the process but it will give you a detailed pattern to follow when importing libraries into your own applications and workbenches. From the **File** menu, select **Import**....



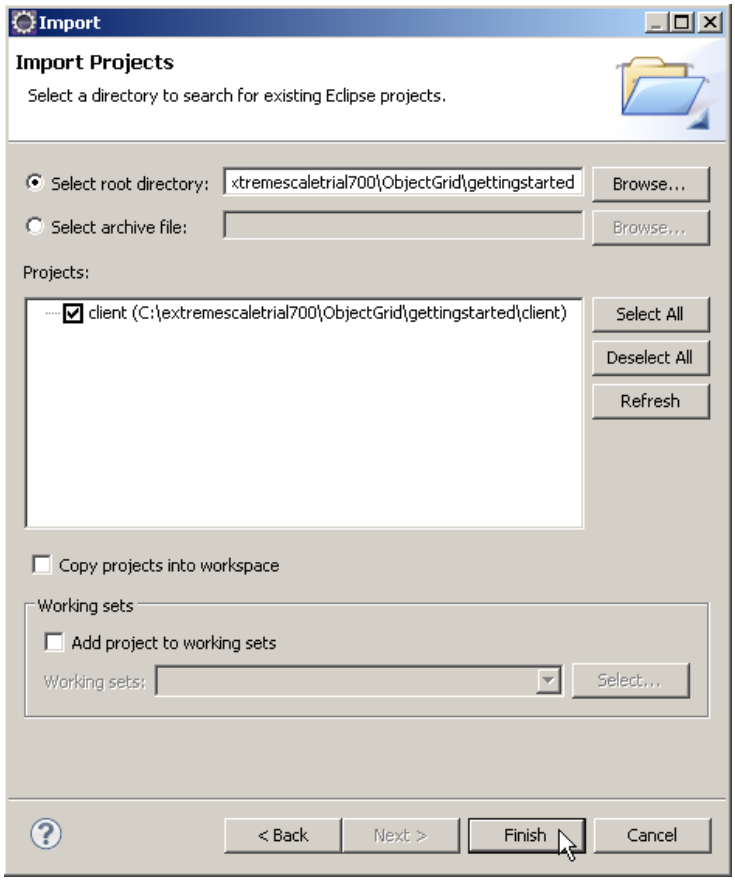
__20. Expand the **General folder. S**elect **Existing Projects into Workspace,** and click **Next**.



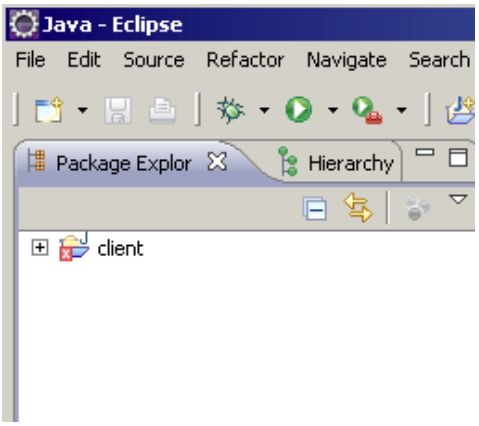
__21. From the Import Project dialog window:

__**a.** In the Select root directory field, Click **Browse…**

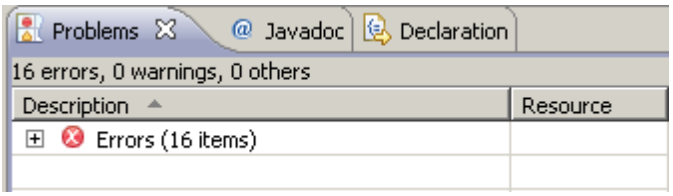
__**b.** Navigate to the folder **C:\extremescaletrial700\ObjectGrid\gettingstarted** and click **OK.**



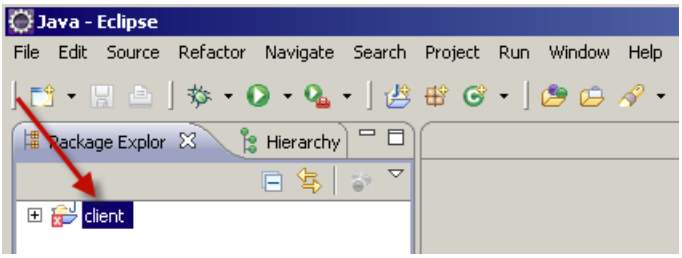
__22. You should now see the client project in your workspace.



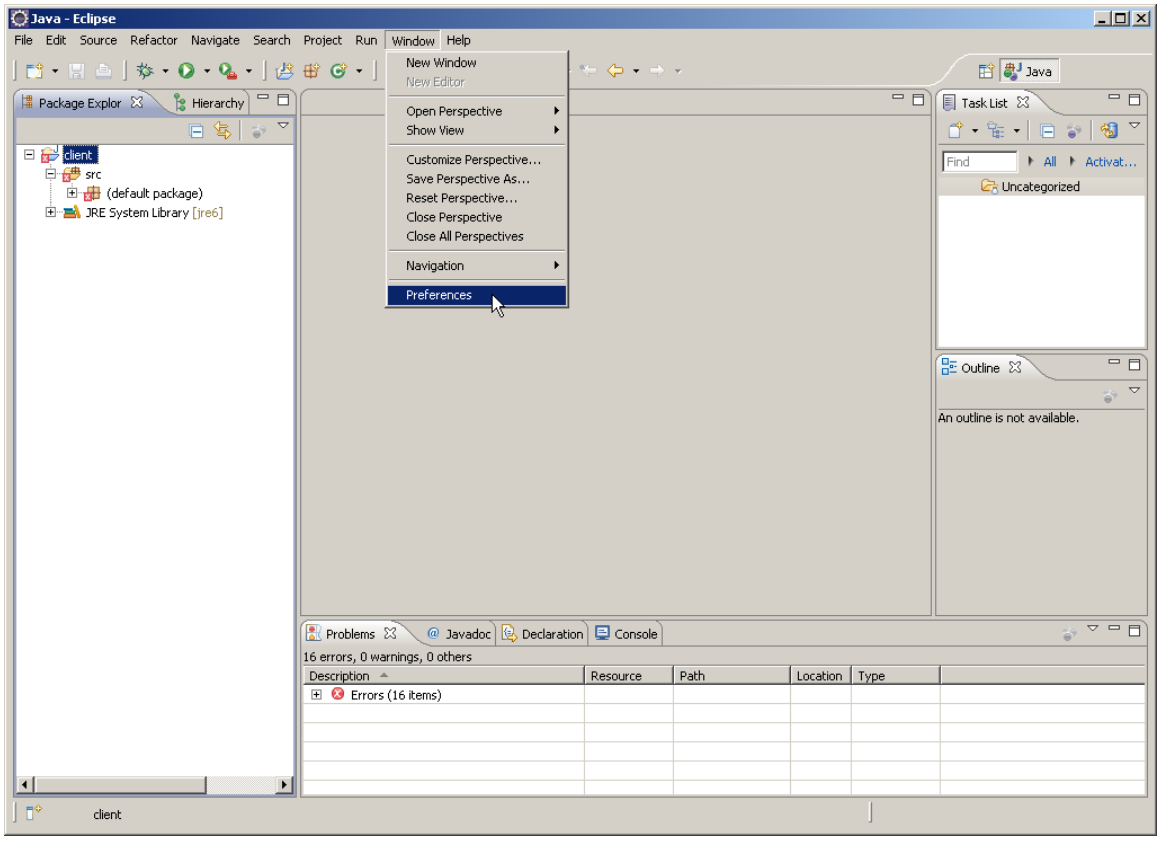
__4. The client project has errors, as shown by the red box beside the client project. Detailed information on the errors is available by examining the Problems tab at the bottom of the perspective. You will now resolve these problems by defining the location of the WebSphere eXtreme Scale client library.



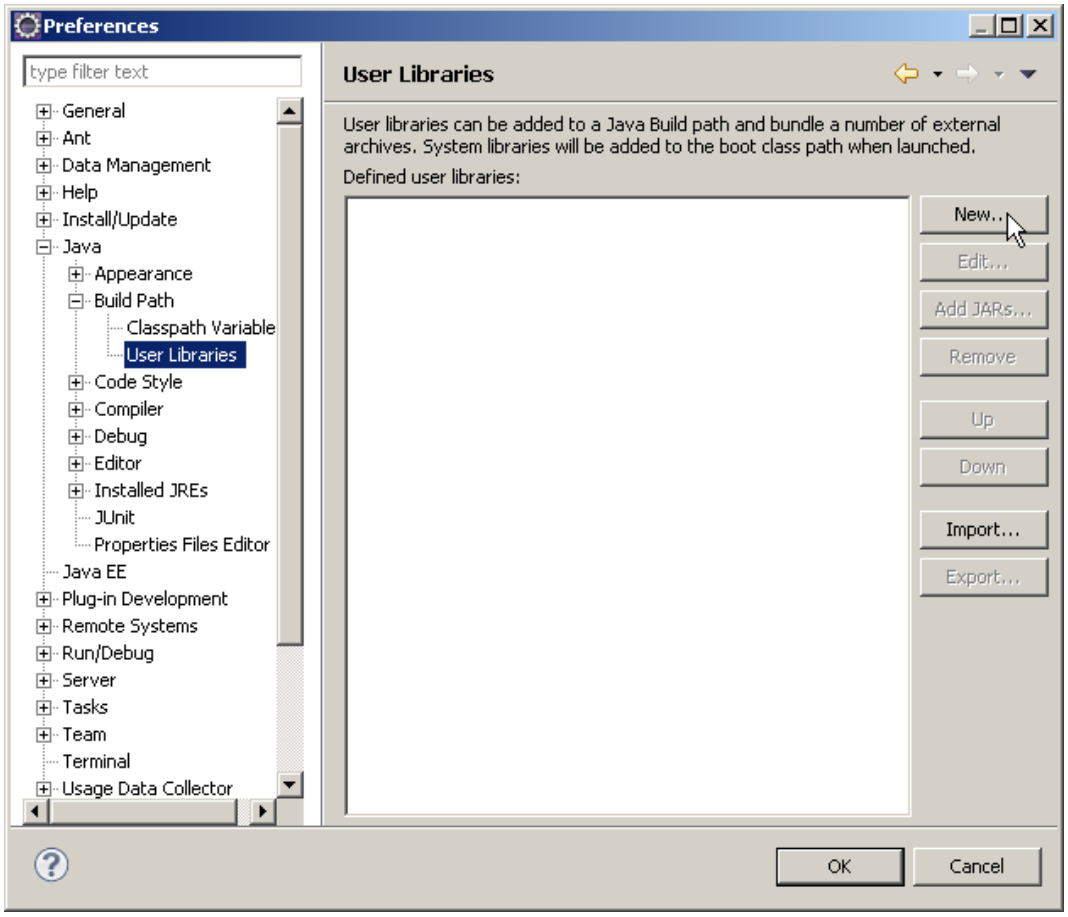
__23. From the Package Explorer view, **select** the client project. The client project selected when the background color of *client* is blue.



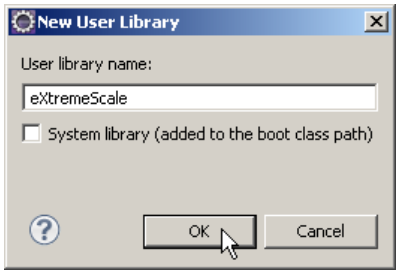
__24. From the **Window** menu select **Preferences...**



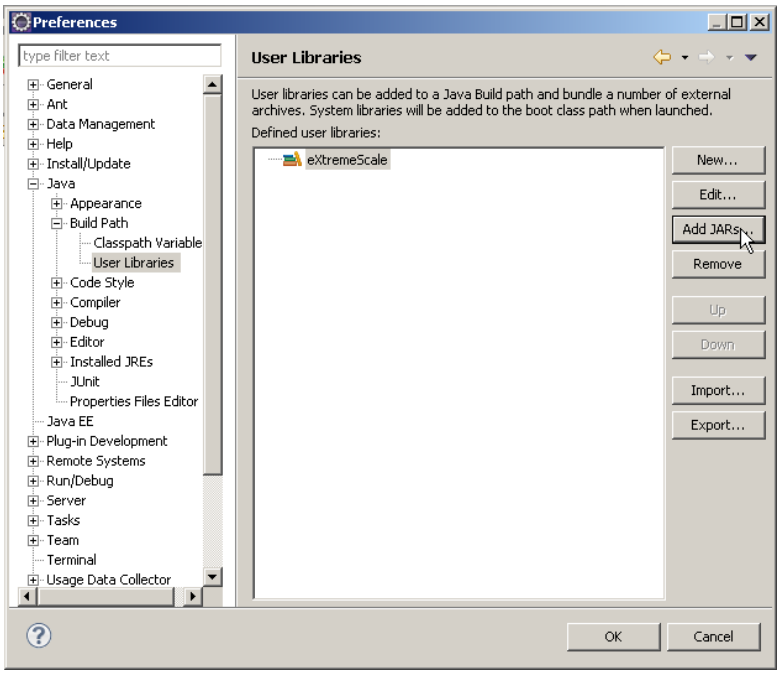
__25. From the left pane, expand **Java→Build Path** and select **User Libraries.** From the workspace, click **New...**



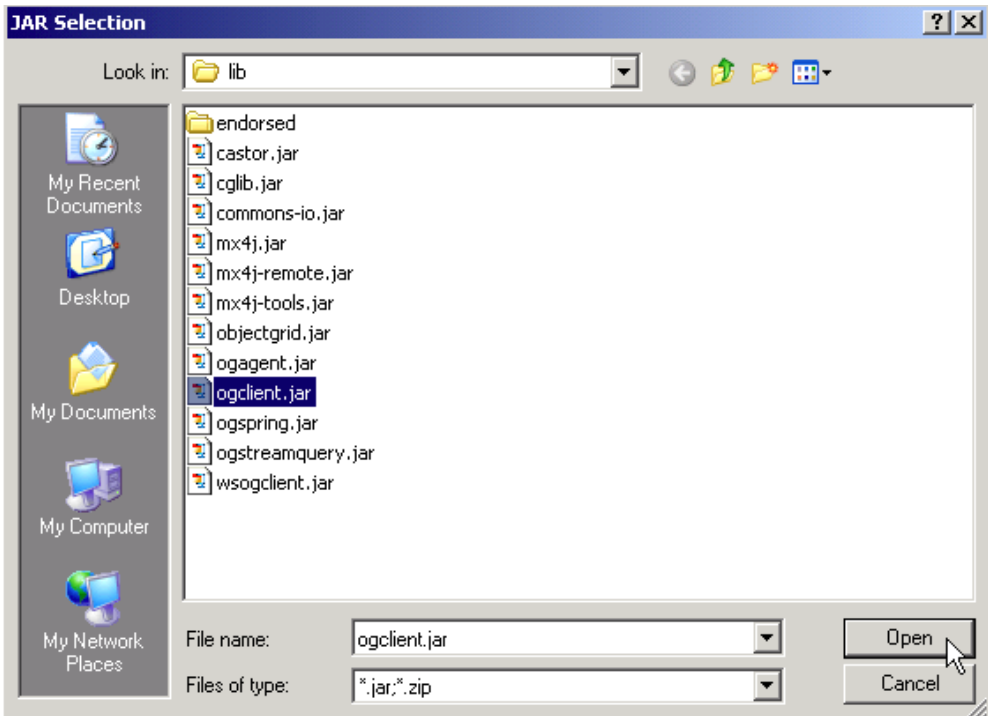
__26. Type **eXtremeScale** in *the **User Library Name*** field and click **OK**.



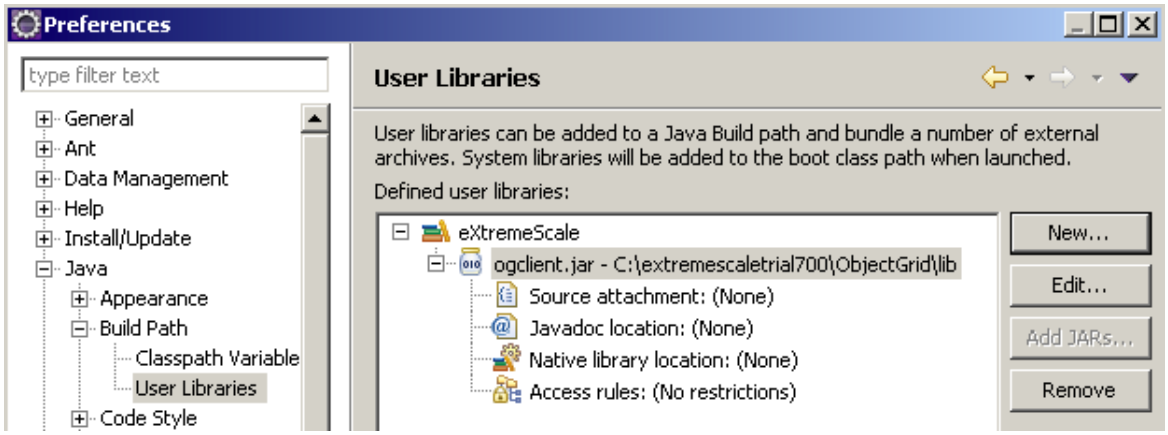
__27. Select the **eXtremeScale** user library. **Click Add JARs...**



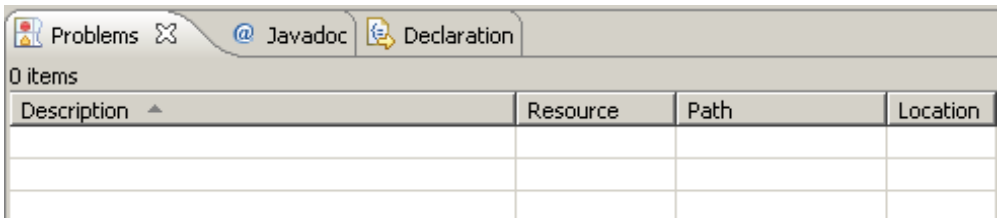
__28. Navigate to **C:\extremescaletrial700\ObjectGrid\lib** and select **ogclient.jar**. Click **Open**.



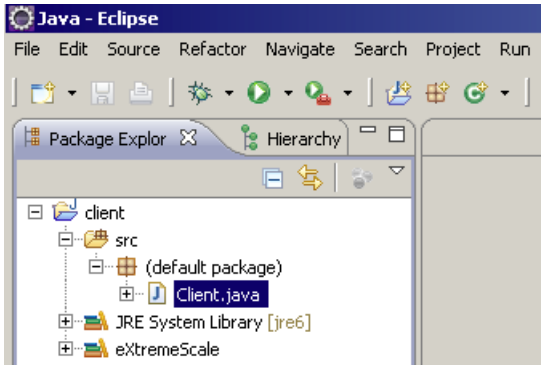
__29. Verify `ogclient.jar` appears in the eXtreme Scale user library definition. Click **OK** to dismiss the Preferences window.



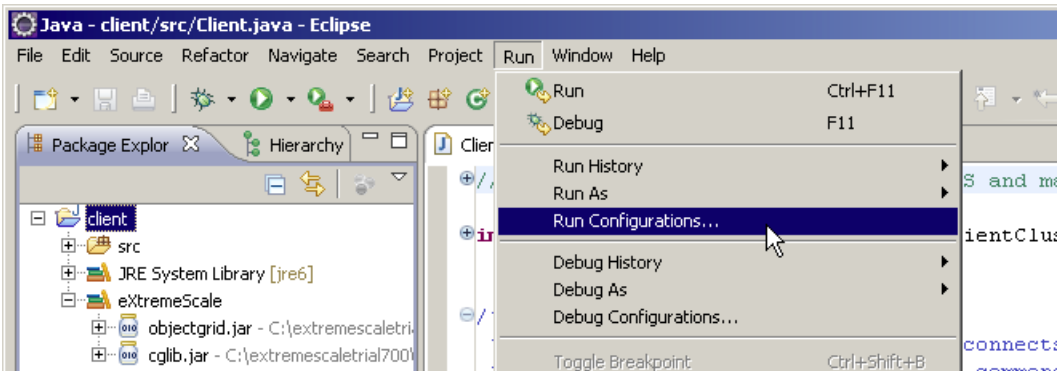
__5. The WebSphere eXtreme Scale client library is now defined and the errors should be resolved under the Problems view.



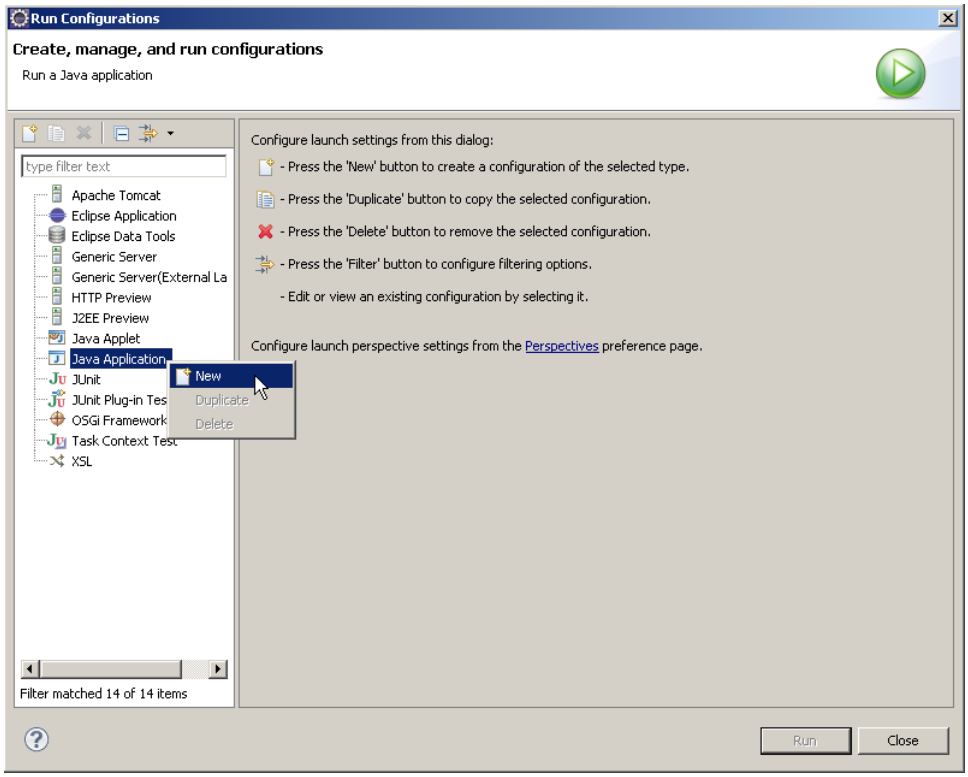
__6. If you are interested in examining the source code used for the client, expand the src folder and locate Client.java. Double-click on Client.java to open it with the Java editor. The getting started client provides a basic sample as to guide to start writing your own grid applications. In this lab, you will not be modifying any of the client source code.



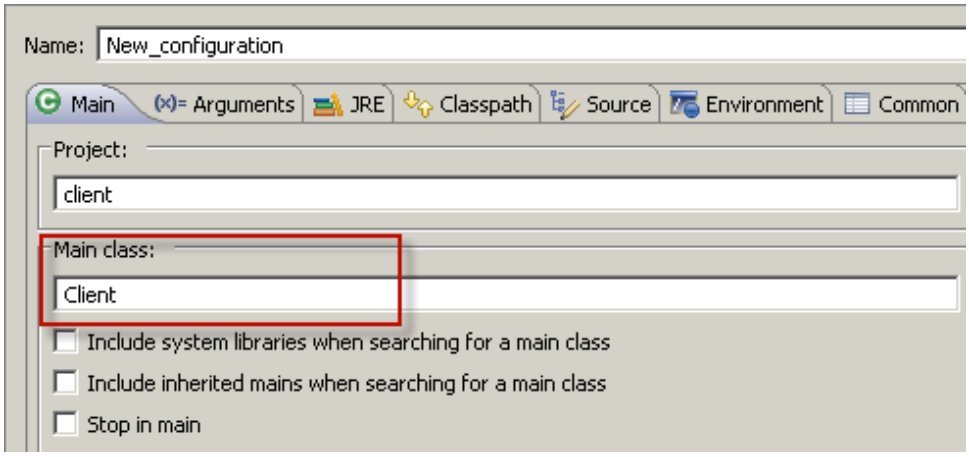
__7. You will now create an Eclipse *Run Configuration*. This allows the client application to run within Eclipse. Select the **client** project. From the **Run** menu, select **Run Configurations...**



__30. Right click on Java Application and select **New...**



__8. In the *Main class:* field, enter **Client**. Click **Apply**.



__31. Click on the **Arguments** tab.

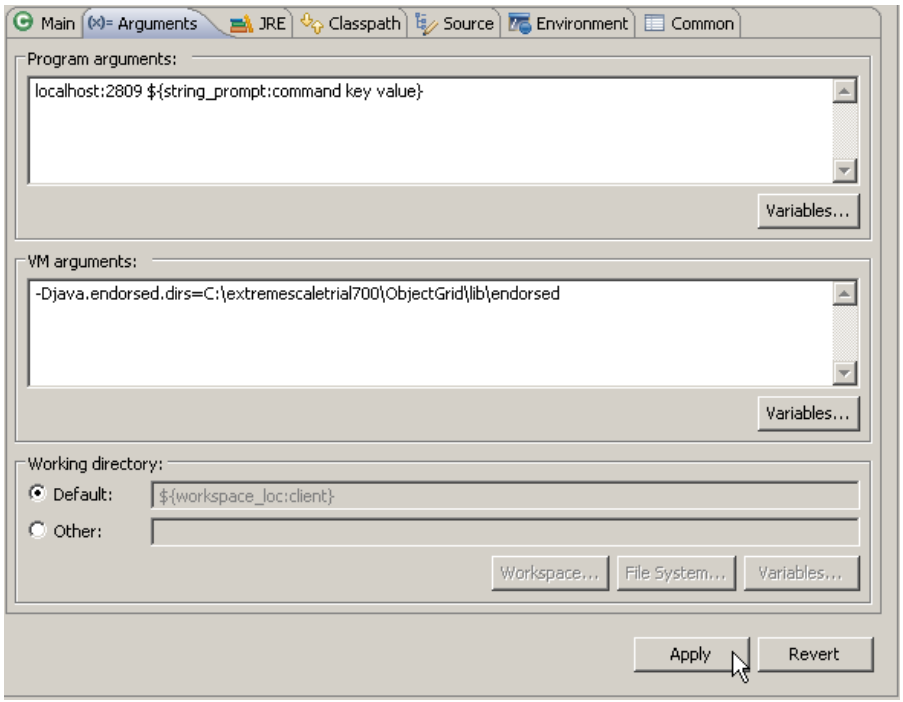
In the Program Arguments section enter the following command :

localhost:2809 ${string_prompt:command key value}

In the VM arguments section enter:

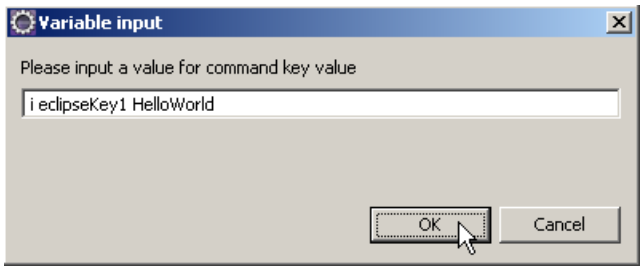
-Djava.endorsed.dirs=C:\extremescaletrial700\ObjectGrid\lib\endorsed

Click **Apply** and then click **Run.**

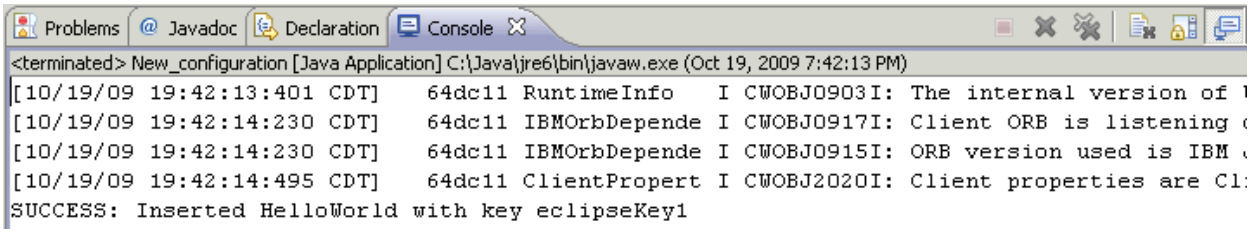


__9. The input window prompts for a command, key, and value separated by spaces. Enter the following:

i eclipseKey1 HelloWorld



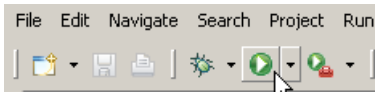
__32. From the Console view, you should see an inserted success message. If you have a failure, ensure your catalog service and container JVMs are still running from the previous section.



You have a functioning WebSphere eXtreme Scale development environment!

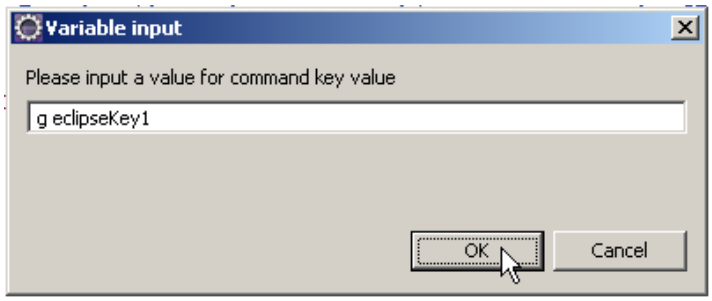
You'll now perform a few additional steps using the client application to see the ease of running the client multiple times.

__33. You can execute the client now by simply clicking the clicking on the green run button as shown below:

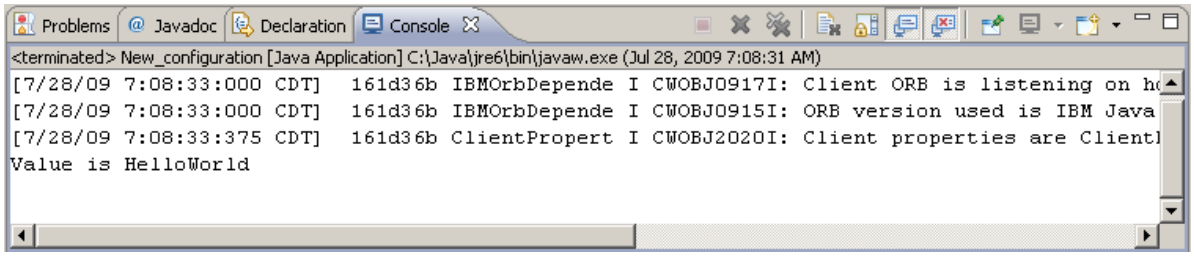


__34. Try clicking the green button. Enter the following get command (notice no value parameter)

g eclipseKey1



The following should appear in the Console view near the bottom of the Eclipse window:



__35. Continue to experiment with the client and enter additional information into the grid. The examples above executed the INSERT (“i”) and GET (“g”) operations. Try experimenting with UPDATE (“u”) and DELETE (“d”).

Make a note of the key/value pairs you insert into the grid. You will access them in the next section. When you are finished experimenting with the client, keep Eclipse running.

Leave
your catalog service and grid JVM's running! You will be looking at the characteristics of the grid you've created in the next section.

1.5 Monitoring a deployed grid

In the final section of this lab, you will become familiar with the basics of WebSphere eXtreme Scale monitoring. You will use the xsAdmin sample utility to provide information on the current layout and specific state of your WebSphere eXtreme Scale topology.

__1. Open a new Command Prompt..

__36. Change to the eXtreme Scale trial bin directory by issuing the command:

cd C:\extremescaletrial700\ObjectGrid\bin

__37. Set the JAVA_HOME system property for this command prompt by issuing the command:

set JAVA_HOME=c:\Java\jre6

__38. As mentioned previously, WebSphere eXtreme Scale distributes key/value data into separate partitions to provide reliability and scalability. You can see the details of the topology using the xsAdmin utility. Issue the following command:

xsadmin.bat -containers

You should see output similar to the following:

```
*** Show all online containers for grid - Grid & mapset - mapSet
Host: think.was7.ibm.com
  Container: server0_C-0, Server:server0, Zone:DefaultZone
    P:0 Primary
    P:1 Primary
    P:10 Primary
    P:11 Primary
    P:12 Primary
    P:2 Primary
    P:3 Primary
    P:4 Primary
    P:5 Primary
    P:6 Primary
    P:7 Primary
    P:8 Primary
    P:9 Primary
  Container: server1_C-0, Server:server1, Zone:DefaultZone
    P:0 SynchronousReplica
    P:1 SynchronousReplica
    P:10 SynchronousReplica
    P:11 SynchronousReplica
    P:12 SynchronousReplica
    P:2 SynchronousReplica
    P:3 SynchronousReplica
    P:4 SynchronousReplica
    P:5 SynchronousReplica
    P:6 SynchronousReplica
    P:7 SynchronousReplica
    P:8 SynchronousReplica
    P:9 SynchronousReplica


  Num containers matching = 2
  Total known containers = 2
  Total known hosts = 1
```

You can see from the xsAdmin output there are two containers cooperating to form the grid. One of the containers holds 13 partitions with the primary copies of the data. The second container also has 13 partitions -- these are replica partitions storing backup copies of the data. Replicas can be synchronous or asynchronous. The types and placement of replicas are determined by WebSphere eXtreme Scale using a deployment policy, which specifies the minimum and maximum number of synchronous and asynchronous replicas.

__39. You can inspect the following two file to get an idea of how this was configured.

C:\extremescaletrial700\ObjectGrid\gettingstarted\xml\deployment.xml

C:\extremescaletrial700\ObjectGrid\gettingstarted\xml\objectgrid.xml

__40. WebSphere eXtreme Scale provides reliable data redundancy and detection of failures. You will now explore these features with your data grid. Shutdown the server0 container JVM by bringing the Command Prompt hosting server0 to the foreground and entering **<Ctrl+C>.** Answer **y** when prompted.

__41. Return to your xsAdmin command prompt and re-issue the command:

xsadmin.bat -containers

You will now see results similar to the output below:

```
*** Show all online containers for grid - Grid & mapset - mapSet
Host: think.was7.ibm.com
  Container: server1_C-0, Server:server1, Zone:DefaultZone
    P:0 Primary
    P:1 Primary
    P:10 Primary
    P:11 Primary
    P:12 Primary
    P:2 Primary
    P:3 Primary
    P:4 Primary
    P:5 Primary
    P:6 Primary
    P:7 Primary
    P:8 Primary
    P:9 Primary


  Num containers matching = 1
  Total known containers = 2
  Total known hosts = 1
```

You can see that now only one container is visible in the grid. Your data is still available and the backup replicas have now been promoted to primaries. If you were to restart the server0 container JVM, WebSphere eXtreme Scale would automatically create new replicas and your data would again be highly available.

.

__42. One last interesting bit of information is the amount of data in the grid and how it is distributed amongst the partitions. Issue the command:

xsadmin.bat -mapsizes

This will produce a report showing each partition and the number of object contained in each.

```
************Displaying Results for Grid - Grid, MapSet - mapSet*****

*** Listing Maps for server0 ***
 Map Name: Map2  Partition #: 2  Map Size: 0  Shard Type: Primary
 Map Name: Map2  Partition #: 11  Map Size: 0  Shard Type: Primary
 Map Name: Map2  Partition #: 7  Map Size: 0  Shard Type: Primary
 Map Name: Map2  Partition #: 10  Map Size: 0  Shard Type: Primary
 Map Name: Map2  Partition #: 8  Map Size: 0  Shard Type: Primary
 Map Name: Map1  Partition #: 12  Map Size: 1  Shard Type: Primary
```

__43. Now use either the command line client or the Eclipse client to access the information you have stored in the grid earlier. Notice the information is still available even though you stopped the container with the primaries.

__2. You will now stop the remaining WebSphere eXtreme Scale servers in a controlled manner. From a command prompt, enter execute the following commands:

xsadmin.bat -teardown server1 – Answer y when prompted.

```
Connecting to Catalog service at localhost:1099
The following servers will be torn down:
  server1

Do you want to tear down the listed servers? (Y/N)

y

Teardown results:

  Server server1 torn down successfully  -  true
```

There is a short wait while this completes.

__44. Stop the catalog service using the command:

xsadmin.bat -teardown cs0

__3. **Close** all Command Prompt windows.

__45. Close Eclipse by selecting File□Exit from the menu.

Congratulations, you have completed the first lab. The getting started sample is provided for a quick introduction to WebSphere eXtreme Scale functionality and basic operation. It consists of shell and batch scripts designed to start a simple grid with very little customization needed. In addition, a client program, including source, is provided to run simple create, read, update, and delete (CRUD) functions to this basic grid. An Eclipse project for the client program is provided.

Important!

Please stop all JVMs used in this exercise when it is complete. They will cause port conflicts and waste memory during later exercise.

Lab 2 Leveraging WebSphere eXtreme Scale to eliminate data access bottlenecks

WebSphere eXtreme Scale is an elastic, scalable, in-memory data grid. It dynamically caches, partitions, replicates, and manages application data and business logic across multiple servers. WebSphere eXtreme Scale installs and deploys in Java EE and Java SE environments.

In the previous lab, you installed stand-alone WebSphere eXtreme Scale in an environment that did not contain WebSphere Application Server or WebSphere Application Server Network Deployment.

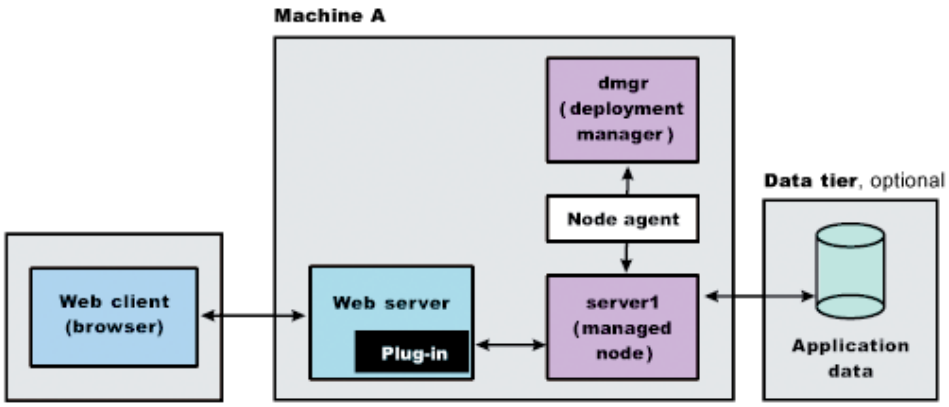
In this lab, you will integrate WebSphere eXtreme Scale Version in an environment that contains an existing configuration of WebSphere Application Server Network Deployment.

Additionally, you will enhance an existing application by applying the capabilities of WebSphere eXtreme Scale. To demonstrate this capability, you will optimize the performance of an application by leveraging WebSphere eXtreme Scale as the intermediary between a database and the application. WebSphere eXtreme Scale can be effectively utilized as a data cache for a database or other data sources, which are generally slower to respond due to data access on a disk storage system.

2.1 Integrate WebSphere eXtreme Scale with WebSphere Application Server

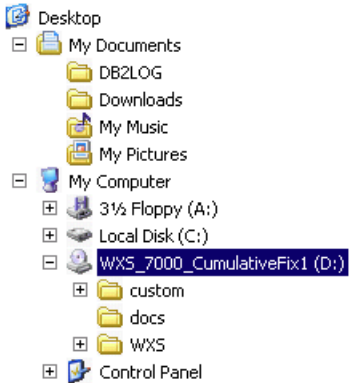
WebSphere Application Server Network Deployment has been pre-installed and configured in the `C:\IBM\WebSphere\AppServer` directory. You will install WebSphere eXtreme Scale Version 7.0 to this location and augment two existing WebSphere profiles to use the WebSphere eXtreme Scale features.

The existing profiles are Dmgr01 (deployment manager) and AppSrv01 (Node Agent and server1). The topology is similar to the diagram below.



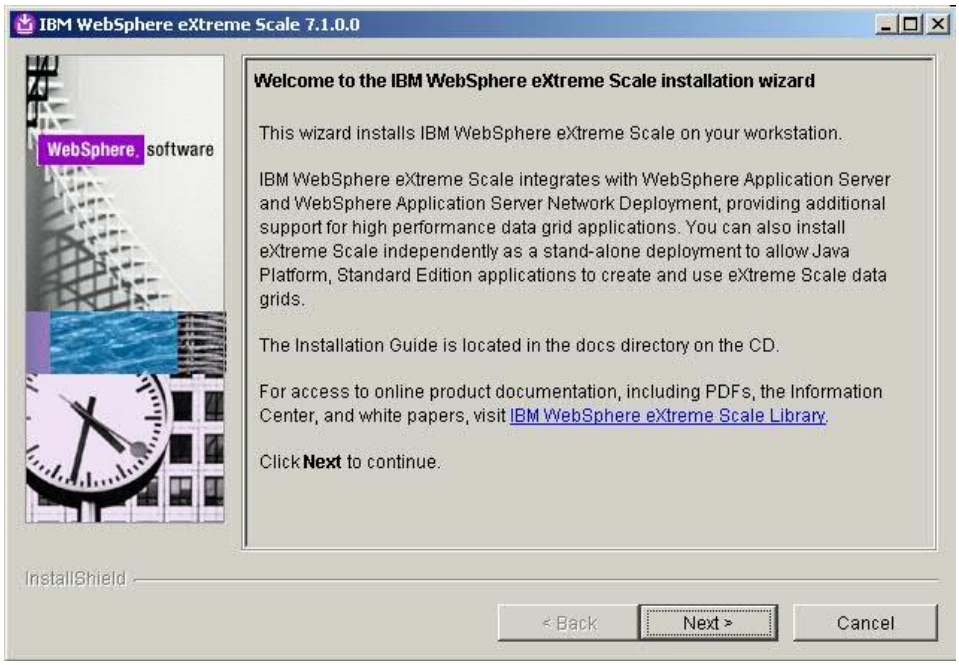
__1. You will use the WebSphere installation wizard to perform the installation. From the Quick Launch toolbar, start **Windows Explorer**

__2. Navigate to the WXS_7000_CumulativeFix1(D:) directory.

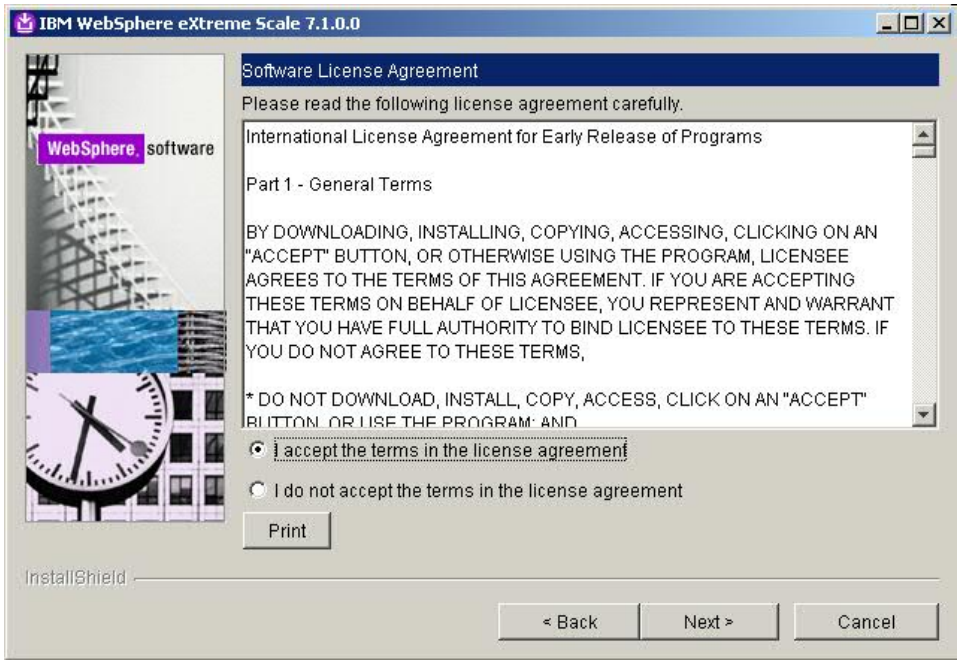


__3. Double-click **`install.bat`** to start the installation wizard.

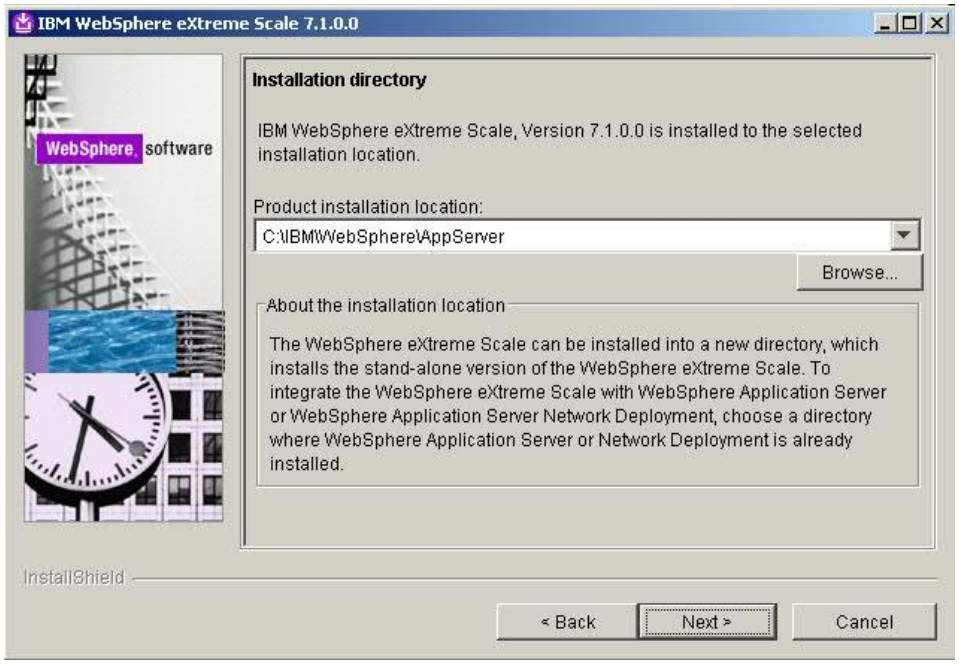
__4. From the Welcome panel, review the introduction and click **Next** to continue.



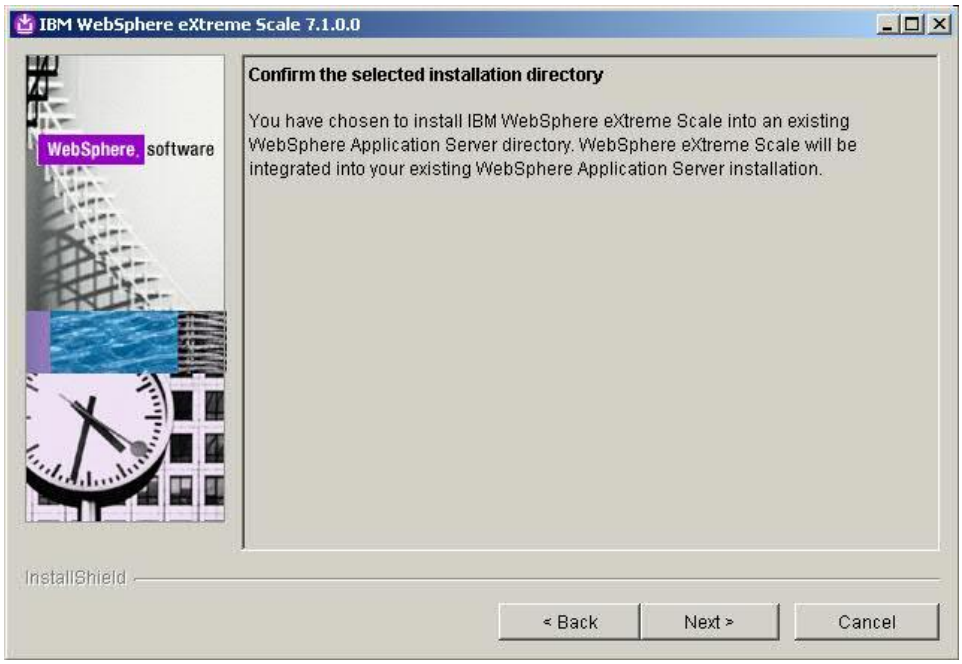
__5. From the Software License Agreement panel, **accept** the license agreement and click **Next.**



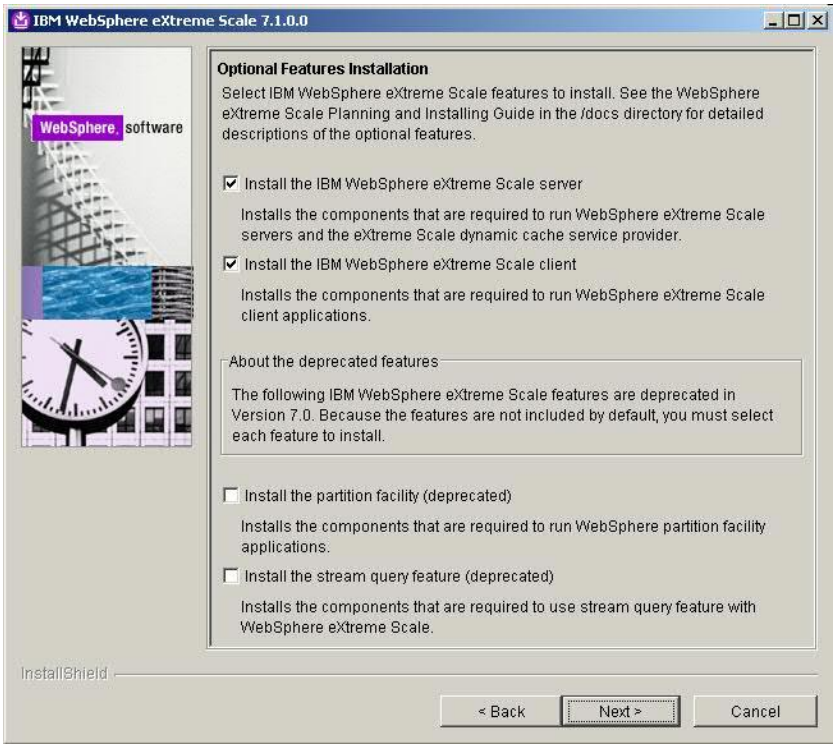
__6. From the Installation Directory panel, accept the location of **`C:\IBM\WebSphere\AppServer`** The installation wizard has detected the existing install of WebSphere Application Server Network Deployment and will install to this directory appropriately. Click **Next**.



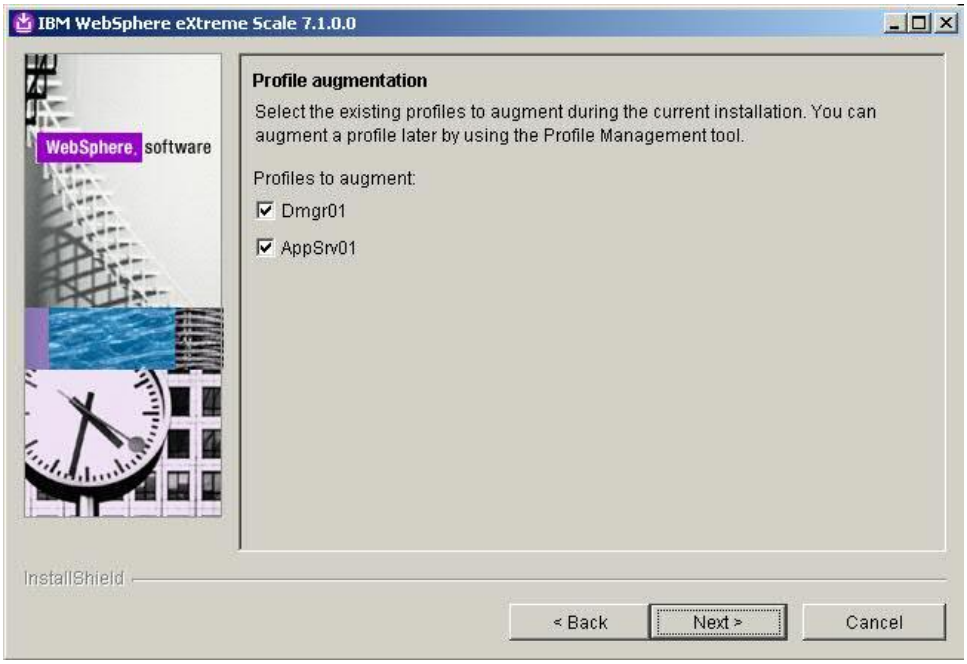
__7. Confirm the installation to an existing WebSphere Application Server directory and click **Next.**



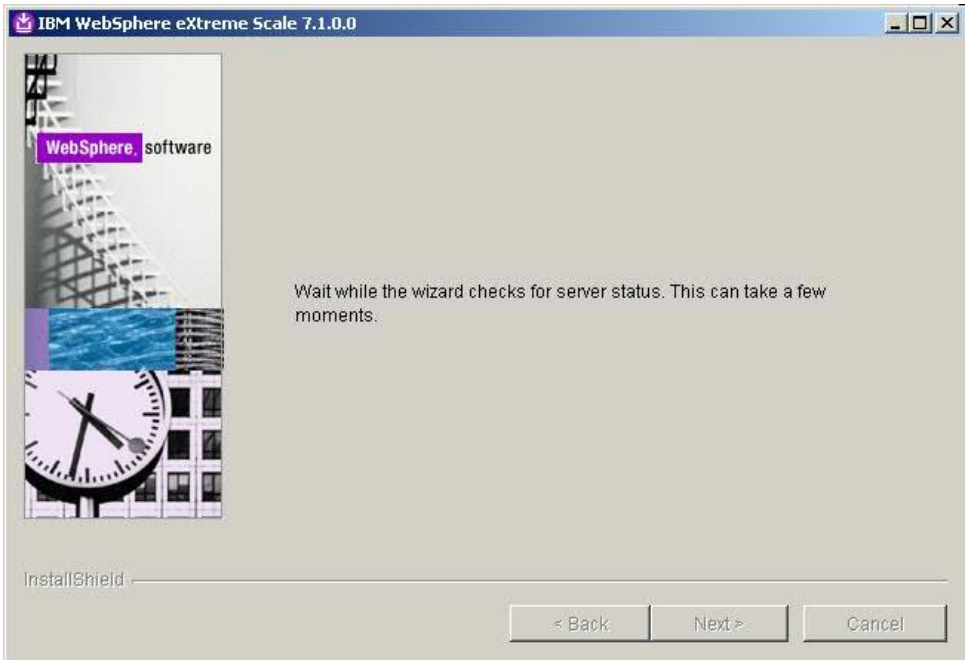
__8. From the Features Installation panel, ensure **Install the IBM WebSphere eXtreme Scale server** and **Install the IBM WebSphere eXtreme Scale client** are selected. The deprecated features will not be used in the lab. Click **Next.**



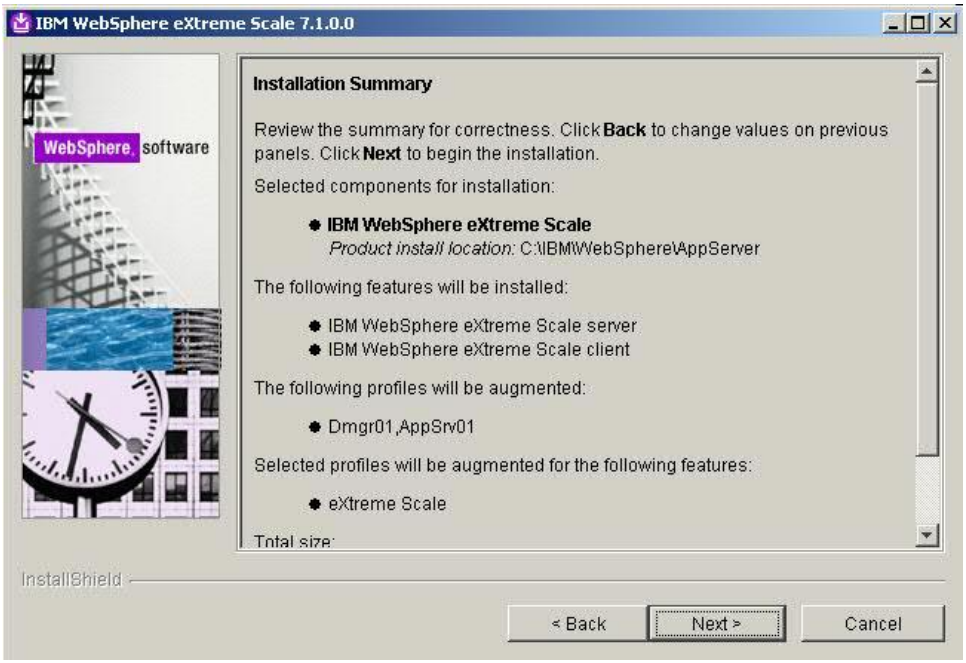
__9. From the Profile Augmentation panel, confirm the **Dmgr01** and **AppSrv01** profiles are selected for augmentation. You must augment any existing profiles that will exploit the WebSphere eXtreme Scale features. If you are running WebSphere Application Server Version 6.1 or Version 7.0, you can also use the graphical Profile Management Tool or the manageprofiles command to augment profiles. Click **Next.**



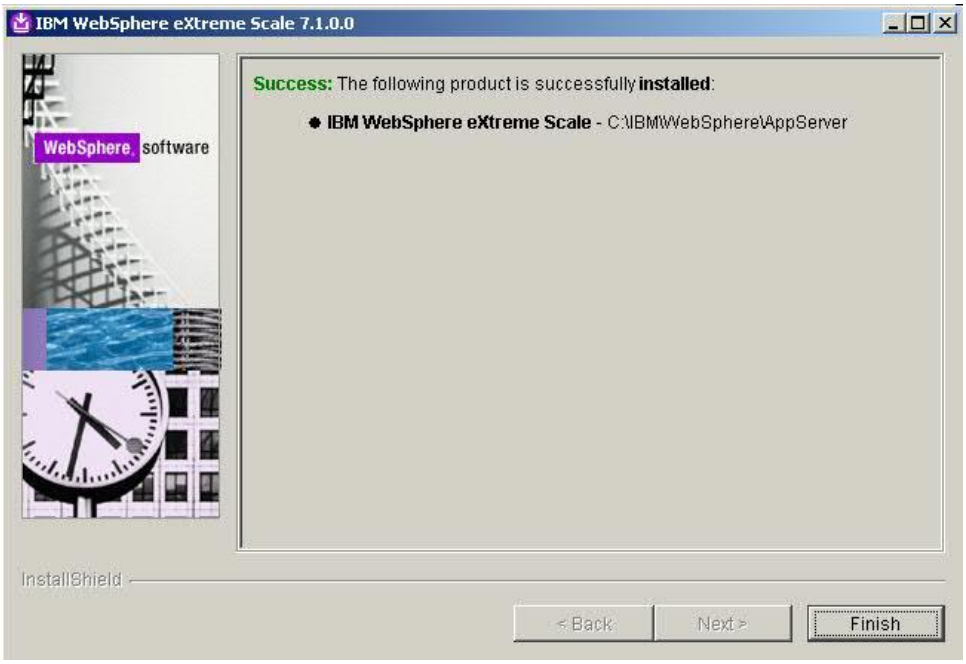
__10. A short wait occurs while the installation wizard checks for any running WebSphere servers.



__11. After the server status check is complete, review the Installation Summary panel and click **Next** to start the installation. The installation will take approximately five minutes to complete.



__12. After the install finishes successfully, click **Finish**.



2.2 Exploring WebSphere eXtreme Scale as an in-line database buffer

With WebSphere eXtreme Scale installed, you will now explore a common challenge that companies encounter and learn how WebSphere eXtreme Scale provides a solution. Here is a description of the problem:

A fictitious online banking Web site with a growing number of users is experiencing slow response times. Their application has data access bottlenecks. They need a way to improve the site performance without upgrading the existing hardware.

WebSphere eXtreme Scale can be easily added to existing environments to save money while improving response time and scale. It eliminates data access bottlenecks by processing requests for data in memory rather than in the database.

This scenario is based on an IBM developerWorks® article -- Leveraging WebSphere Extreme Scale as an in-line database buffer:
http://www.ibm.com/developerworks/websphere/library/techarticles/0906_vuong/0906_vuong.html

2.3 Preparing the environment

You will begin by examining the performance characteristics of the existing bank application. The data model for this scenario is a User which contains many User Accounts and many User Transactions. The application reads and writes the User information to a database. The database used in this lab is IBM DB2® Universal Database™.

__1. From the Quick Launch toolbar, start a **Command Prompt.** You are placed in the **`c:\IBM\WebSphere\AppServer`** directory.

__2. The database needs to be populated with the User, User Account, and User Transaction information. From the **Command Prompt**, enter the following command:

```
cd c:\ClassMaterial\Lab2
```

__3. Enter the following command to populate the database with 1000 generated users and their associated accounts and transactions:

populateDB.bat

You can watch the progress of this command as it creates the users.

```
78  userPUDB2  INFO   [Thread-3] openjpa.Runtime - Starting OpenJPA 1.2.3-SNAPSHOT
persist users from 0 to 1000. totalUsers=1000
@0
@100
@200
@300
@400
@500
@600
@700
@800
@900
User between 0 and 1000 are persisted directly
```

2.4 Start the WebSphere Application Server Network Deployment environment

You have installed WebSphere eXtreme Scale in a WebSphere Application Server Network Deployment environment and augmented two existing WebSphere profiles. You will now start the servers in this environment and examine the bank application configuration.

__1. Enter the following commands from the Command Prompt to start the Deployment Manager

__a. cd c:\IBM\WebSphere\AppServer\profiles\Dmgr01\bin

__b. startManager.bat

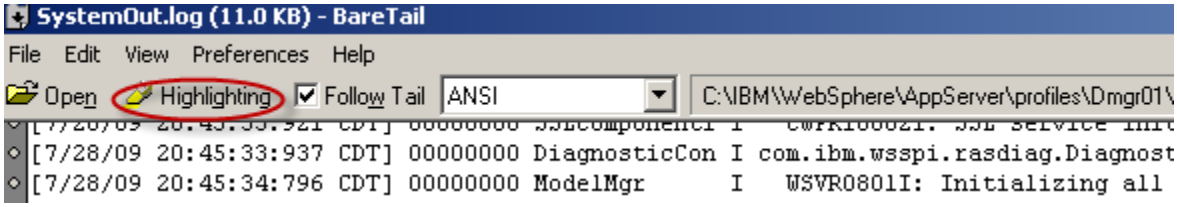
Do not wait for the Deployment Manager startup to finish, continue to the next step.

__2. Monitor the Deployment Manager startup using BareTail. BareTail is a Windows equivalent of the UNIX® tail -f command, allowing one to view a growing file.

__a. From the Windows Taskbar, select **Start → Programs → BareTail**

__b. BareTail has been configured to monitor the *SystemOut.log* file of the Deployment Manager.

__c. You will highlight a string in the SystemOut.log file to examine the details of WebSphere eXtreme Scale during the startup process. Click **Highlighting**



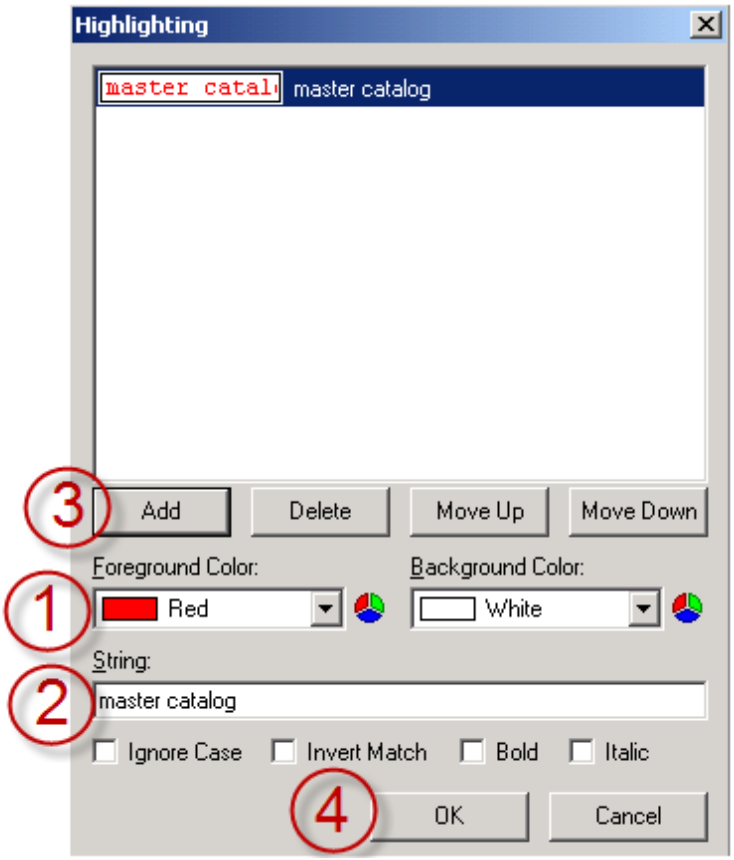
__d. In the Highlighting dialog:

__i. Select a *Foreground Color* of **Red**

__ii. In the *Sting* textbox, enter **master catalog**

__iii. Click the **Add** button

__iv. Click the **OK** button



__e. In the log file text, scroll up and locate the text highlighted in red. These messages show the WebSphere eXtreme Scale catalog service starting in the Deployment Manager.

```
RouterImpl     W   CWOBJ2002W: No available routing table for this replication group 1.
SysAdminServi  I   CWOBJ1914I: System administration network service is started.
SysAdminServi  I   CWOBJ1915I: System administration handler is started.
ClientNetwork  I   CWOBJ1901I: Client server remote procedure call service is started.
ClientNetwork  I   CWOBJ1902I: Client server remote procedure call handler threads are started.
CatalogServic  I   CWOBJ8102I: Notify that master catalog service is created with domain= thinkCell01
BaseMap        I   CWOBJ0033I: Class, class com.ibm.ws.objectgrid.IDLBindInfoImpl, does not implement
CatalogServic  I   CWOBJ8106I: The master catalog service cluster activated with cluster CatalogClust
BaseMap        I   CWOBJ0033I: Class, class com.ibm.ws.objectgrid.container.IDLObjectGridServerInfoIm
```

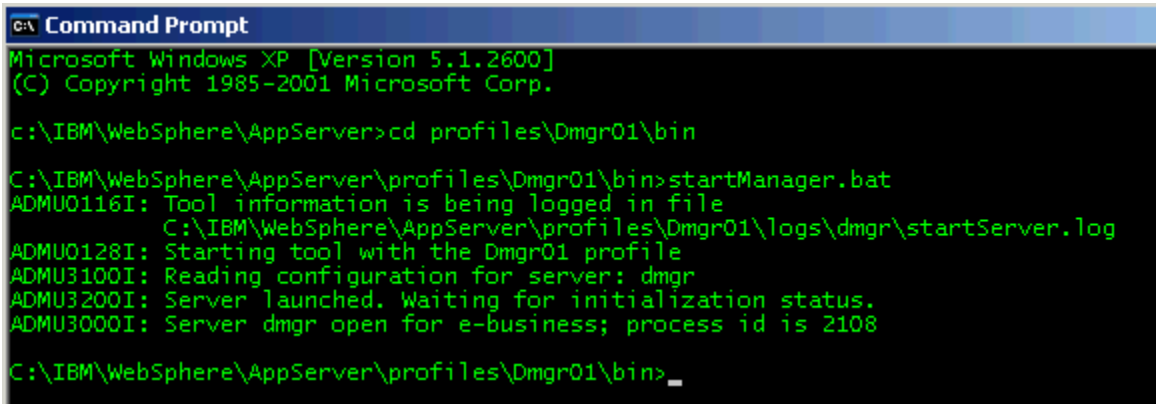
The catalog service acts as a coordinator for the other servers participating in the grid and manages topology information. With WebSphere Application Server Network Deployment, the catalog service process runs in the Deployment Manager by default, but you can configure it to run in a Node Agent or another application server process.

Important!

A single, non-clustered catalog service is acceptable for development environments. For a production environment, you should use a clustered, highly-available catalog service.

__f. **Close** the BareTail utility

__g. The Deployment Manager is started when the `Server dmgr open for e-business` message is displayed in the Command Prompt window.



__3. Start the Node Agent using a procedure similar to starting the Deployment Manager:

__a. From the Command Prompt, enter the following commands:

__i. cd ..\..\AppSrv01\bin

__ii. startNode.bat

__b. The Node Agent is started when the `Server nodeagent open for e-business message` is displayed.

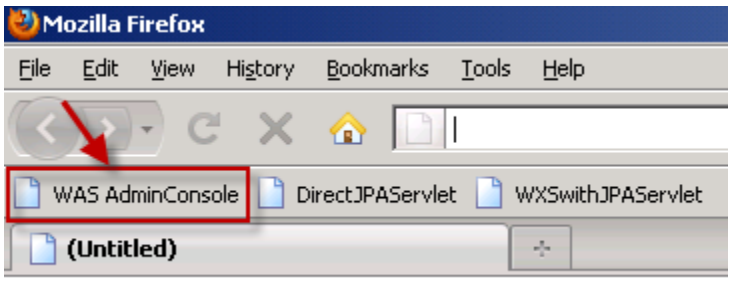
__c. **Close** this Command Prompt.

2.5 Review the WebSphere Application Server configuration

The WebSphere environment and configuration needed to run the bank application is pre-configured. You will take a quick tour of the important settings.

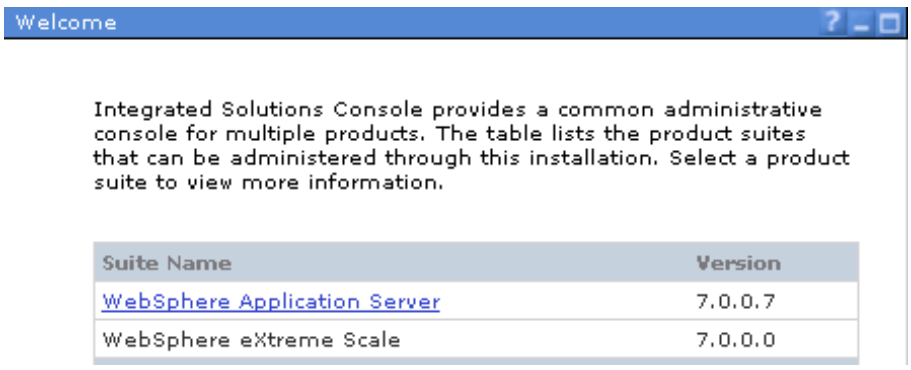
__1. **Start** Mozilla Firefox or Internet Explorer. There are shortcuts for both browsers on the Windows desktop.

__46. **Connect** to the WebSphere Integration Solutions Console (also known as the WAS AdminConsole) using the URL http://think:9060/ibm/console or use the bookmark.

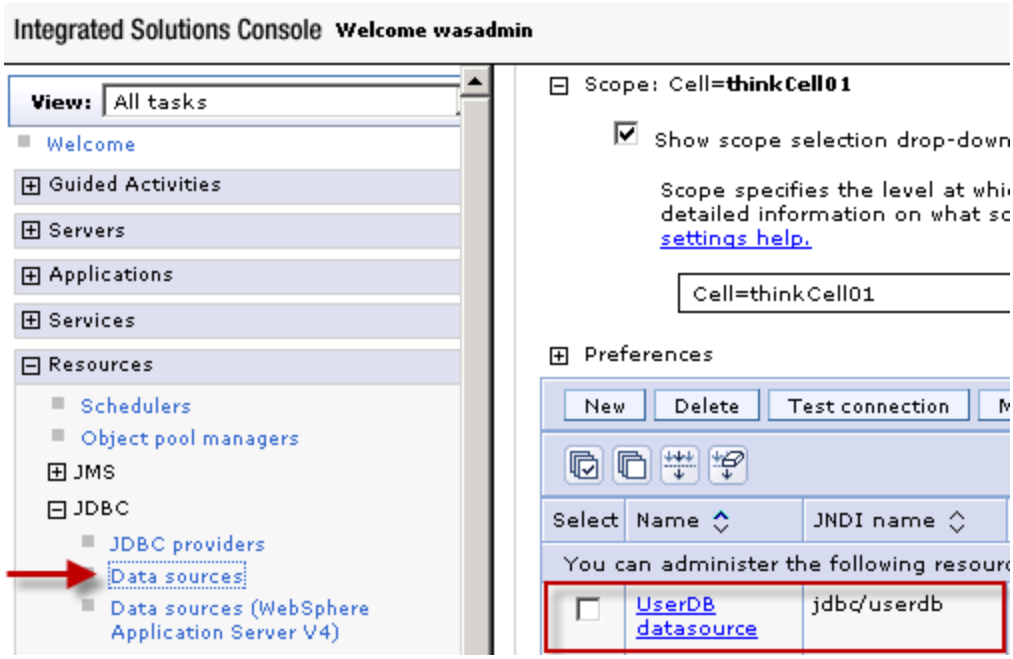


__4. Login with userid **wasadmin** and password **wasadmin**.

__5. From the Welcome panel, notice that WebSphere eXtreme Scale is installed.

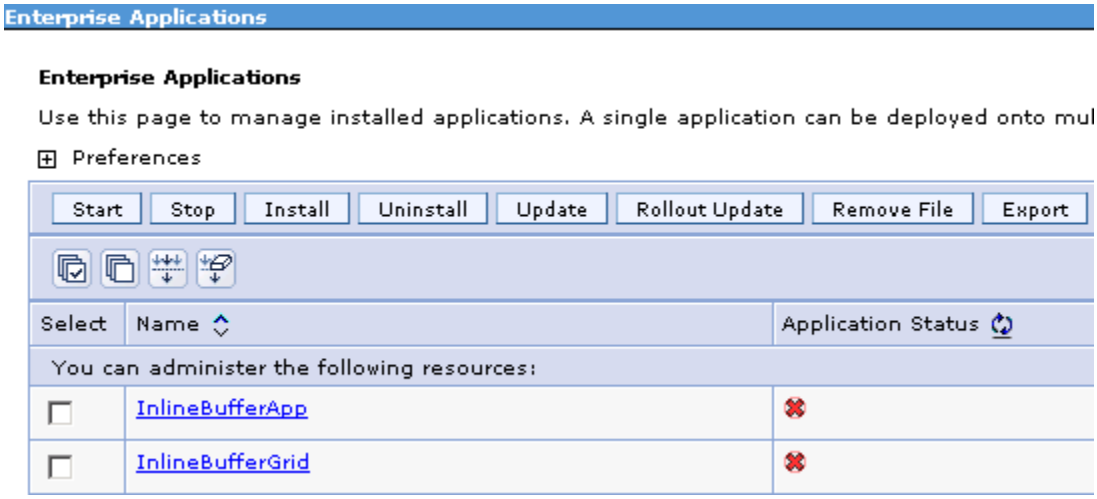


__6. The bank application uses the Java Persistence API (JPA) specification as a mapping between Java Objects and relational databases. The application requires a JDBC™ data source to access the DB2 database. From the navigation tree, expand **Resources → JDBC** and click **Data sources.** In the workspace**,** you will see the UserDB datasource.



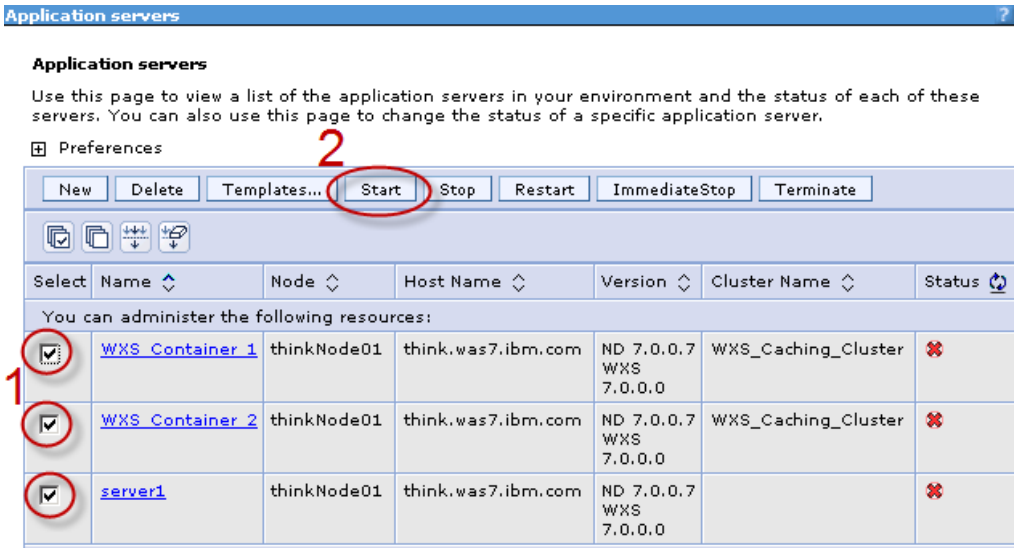
__7. The applications used for this scenario have been pre-installed. From the navigation tree, expand **Applications → Application Types** and click **WebSphere enterprise applications.**

__8. In the workspace**,** you will see two applications. The *InlineBufferApp* application runs the business logic to retrieve and store user information. The *InlineBufferGrid* application is a WebSphere eXtreme Scale application. It does not execute any business logic; it acts as a data grid.

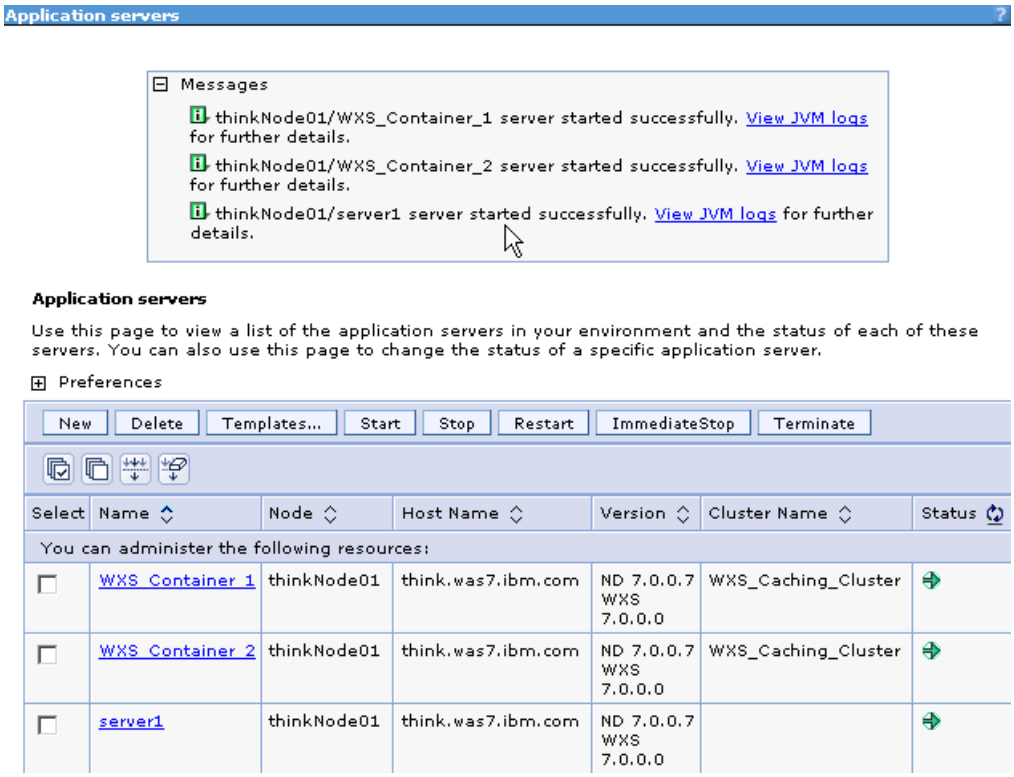


__9. Now start the application servers hosting the enterprise applications used for this scenario. From the navigation tree, expand **Servers → Server Types** and click **WebSphere application servers**.

__10. From the workspace, select **all** the servers and click the **Start** button.



__11. A successful start message and green status icons appear when startup is complete.

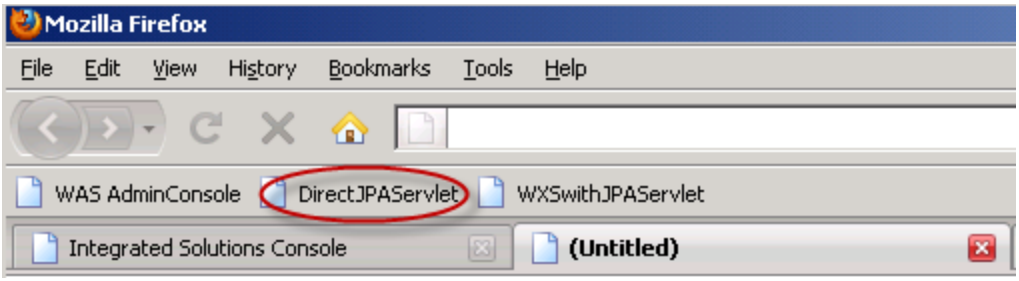


2.6 Performance profile of the bank application

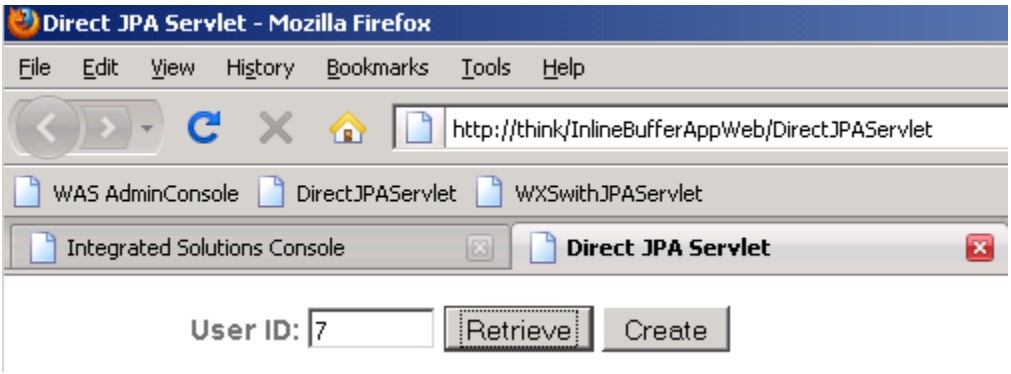
For analyzing and measuring the performance of the bank application, you will start by using a browser for basic observations.

__1. In your current browser, open a new, blank tab by entering **CTRL-T**. Both Mozilla Firebox and Internet Explorer support tabbed browsing. You'll interact with the bank application in its own tab.

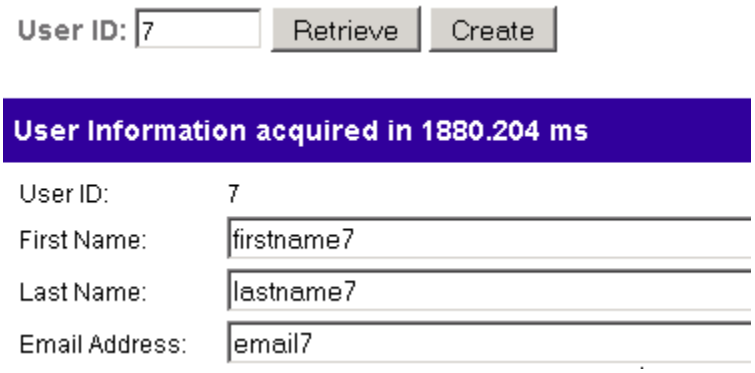
__2. Click on the **DirectJPAServlet** shortcut. This displays the existing bank application which uses the Java Persistence API (JPA) to directly access the DB2 database.



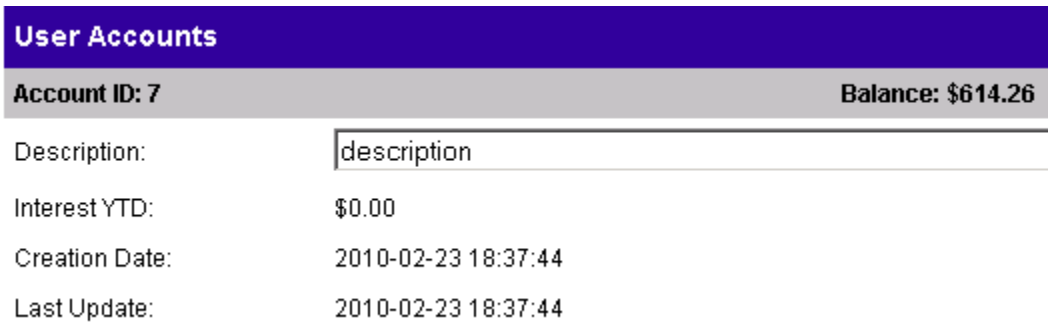
__47. Enter a userid between 0 and 999. Click **Retrieve**. This example uses User ID 7.



__48. The User information is displayed, along with the time required to retrieve the information. The first access is a bit sluggish.



__49. Below the User ID information, the User Accounts are listed. Approximately fifteen User Accounts are created for each User. There is a OneToMany relationship between Users and User Accounts.



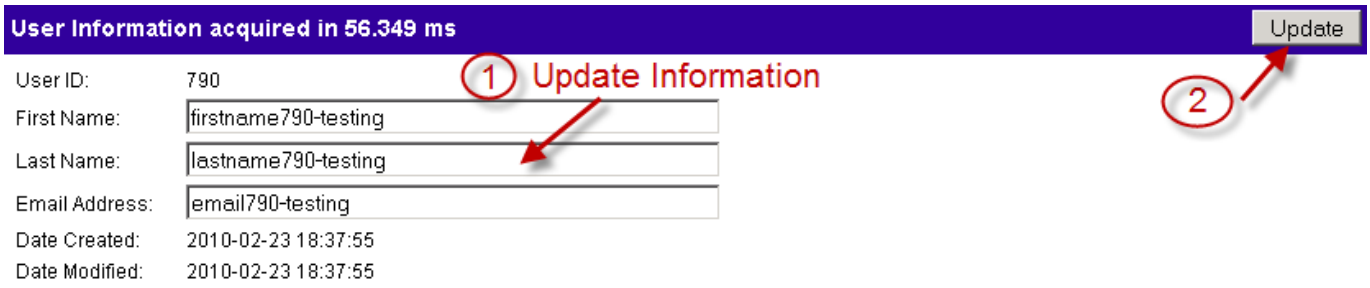
__12. The User Transactions are listed next. There is a OneToMany relationship between User Accounts and User Transactions. You can see that a single bank User has a meaningful amount of data associated with it, and it is expensive to retrieve.

__50. From the top of the page, click the **Retrieve** button again.

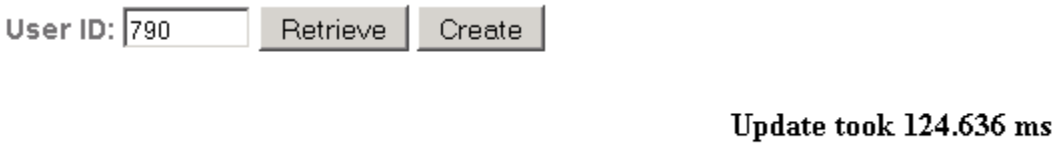
__51. The response time will significantly improve.

__52. Enter a different userid and click the **Retrieve** button again. In this example, the retrieve took 45 milliseconds to complete.

__13. Update the First Name, Last Name and Email Address of the User. Click **Update**.



__14. The update time is shown on the right. In this example it took ~125 milliseconds.



__15. As a rough performance baseline, the direct JPA database application required 45 milliseconds for reads and 125 milliseconds for writes.

2.7 Performance profile of the bank application using WebSphere eXtreme Scale as an in-line data buffer

In this section of the lab, you will investigate the performance of the bank application when it leverages WebSphere eXtreme Scale as the intermediary between the database and the application. WebSphere eXtreme Scale is easily added to existing applications, improving response time and scale. It eliminates data access bottlenecks by processing requests for data in memory rather than in the database.

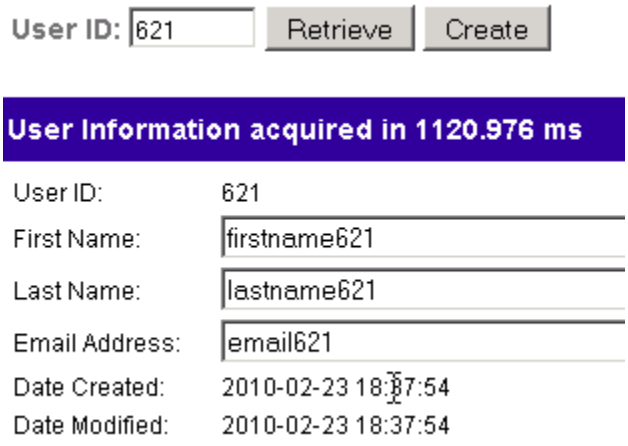
The architecture of the application is now:

- The application checks to see if WebSphere eXtreme Scale contains the desired data.
- If the data is there, the data is returned to the application. If the data is not there, the data is retrieved from the back-end by WebSphere eXtreme Scale so that the next request can use the cached copy.
- Changes are written to the cache and back-end synchronously and transactionally. A *write-through* cache.

__1. From the browser, click the **WXSwithJPAServlet** shortcut. This accesses an application that uses the Java Persistence API (JPA) together with WebSphere eXtreme Scale to access the DB2 database. WebSphere eXtreme Scale provides built-in support for applications leveraging Object to Relational Mapping (ORM) specifications such as JPA and Hibernate.

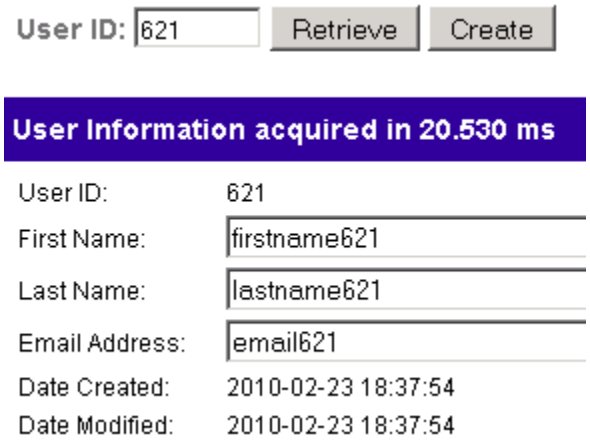
__2. Enter a userid and click **Retrieve**

__3. The access time to acquire the User Information is displayed.



As you have seen before, the first access time is a sluggish. However, the information for User 621 is now cached in WebSphere eXtreme Scale. Subsequent requests can use the cached copy.

__4. Click the **Retreive** button again. This has an exceptionally fast response time since the information was retrieved from the WebSphere eXtreme Scale cache and did not access the database.



Databases such as DB2 provide exceptional performance and utilize caching, using features such as bufferpools to avoid accessing the disk.

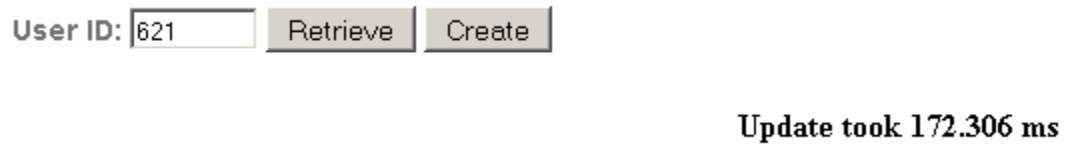
How is WebSphere eXtreme Scale providing additional value in this scenario?

Databases provide data in SQL form and for many applications this data requires a transformation, mapping from a relational model to an object model. This can be slow. WebSphere eXtreme Scale caches data in the native application form, which makes it significantly faster to fetch information from the cache rather than fetching the information from a database. It reduces the path length on the application side because there is no object to relational mapping.

In addition, the database will typically be in a separate tier. WebSphere eXtreme Scale will offload the database, reducing redundant calls and processing. The effectiveness of the cache is directly proportional to the **hit ratio**, which is the percentage of requests satisfied by having the item in the cache.

__5. Modify First Name, Last Name, and Email Address for the User and click **Update**

__6. The update time is shown on the right.



The update time is typically higher than going to the database directly, since the WebSphere eXtreme Scale transaction updates both the cache and the DB2 database.

Another important factor in determining what data to cache is the write-to-read ratio. Caching works best when the data does not change often. For example, if user profiles were cached, they change infrequently and so the write-to-read ratio is small.

__7. You can retrieve and update additional User IDs to further explore the characteristics of using WebSphere eXtreme Scale as a front-end cache for a database to increase throughput while reducing database load and contention.

By default all updates to the grid are written synchronously to the backend data source as part of the transaction so the cache is never out of sync with the database (a ‘write-through’ configuration). Optionally, the grid can be configured to buffer changes to the database for some period of time before asynchronously grouping all of the changes into a batch transaction and sending them in one large transaction (a ‘write-behind’ configuration).

2.8 Monitoring performance with WebSphere Application Server PMI

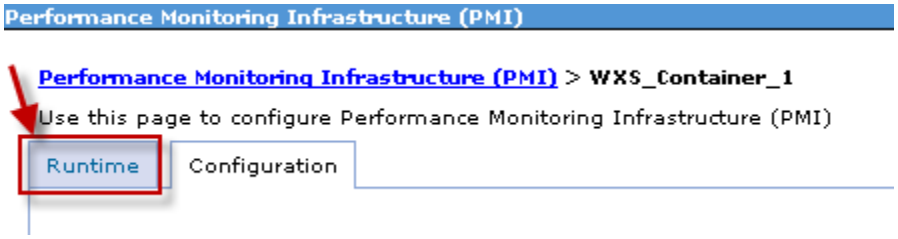
WebSphere eXtreme Scale supports Performance Monitoring Infrastructure (PMI) when running in a WebSphere Application Server or a WebSphere Extended Deployment application server. PMI collects performance data on runtime applications and provides interfaces that support external applications to monitor performance data. You can use the administrative console or the wsadmin tool to access monitoring data.

WebSphere eXtreme Scale can also be monitored using several popular enterprise monitoring solutions. Plug-in agents are included for IBM Tivoli® Monitoring and Hyperic HQ, which monitor WebSphere eXtreme Scale using publicly accessible management beans. CA Wily Introscope uses Java method instrumentation to capture statistics.

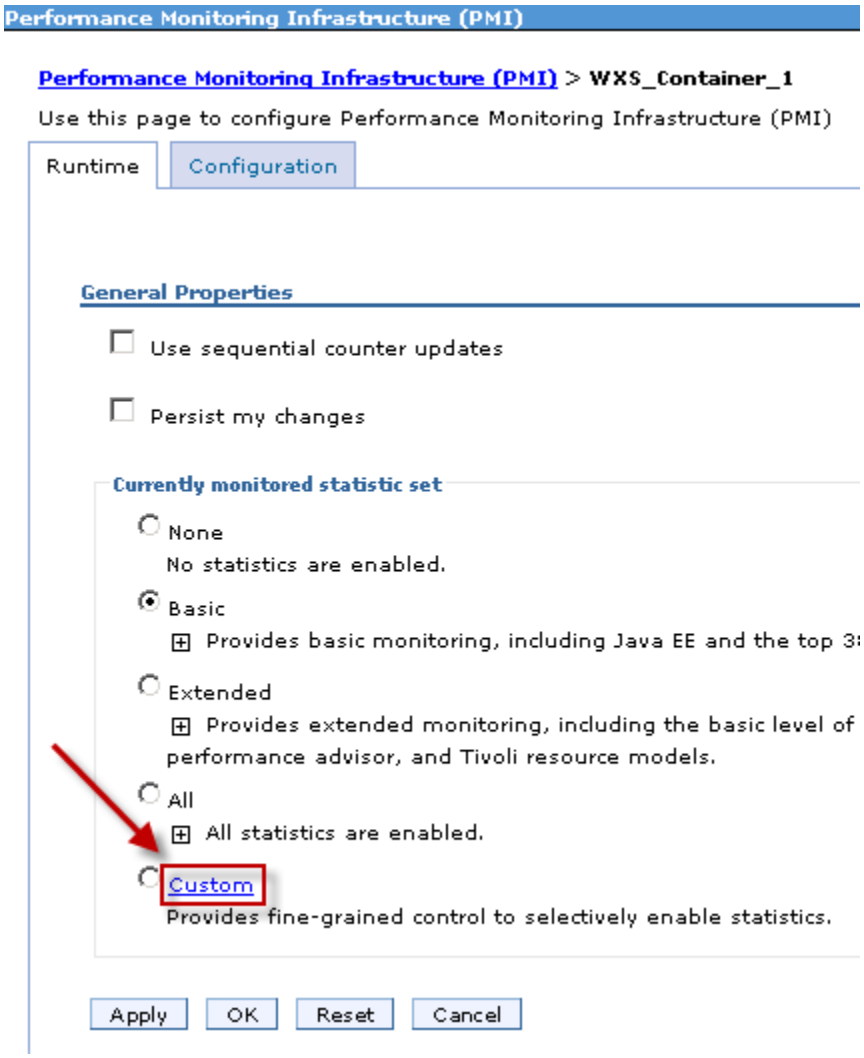
__1. WebSphere eXtreme Scale performance monitoring is not enabled by default, you will enable it now. From the administrative console, expand **Monitoring and Tuning** and select **Performance Monitoring Infrastructure (PMI)**

__**2.** In the workspace, select **WXS_Container_1**

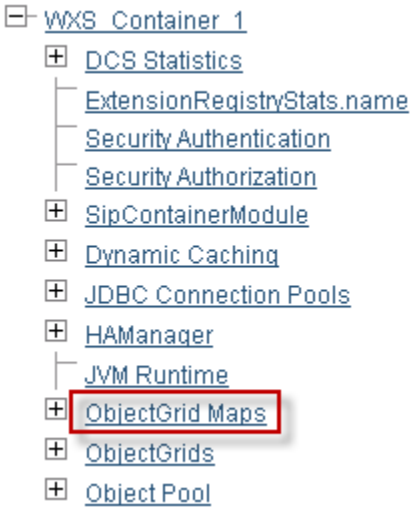
__3. Click the **Runtime t**ab



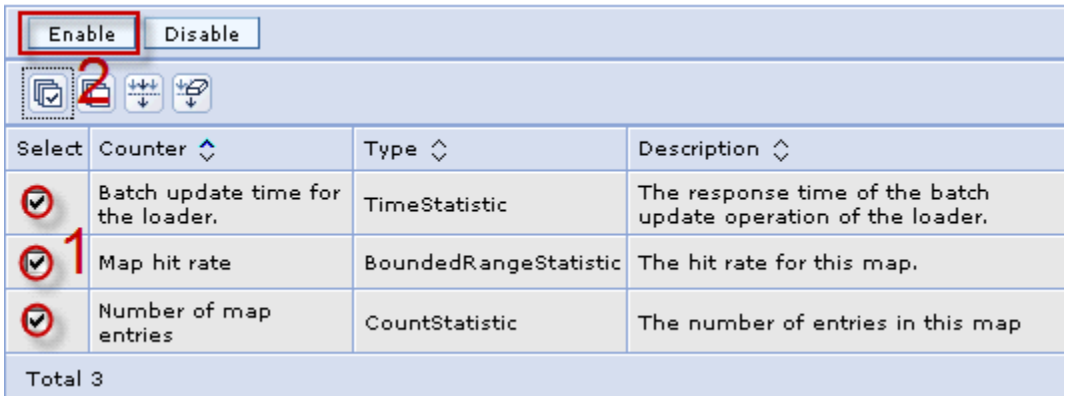
__4. Click the **Custom** link



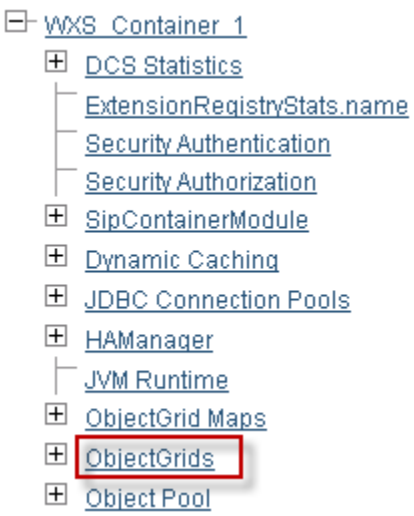
__5. Click **ObjectGrid Maps**



__6. In the workspace, **select** all the counters, then click **Enable**

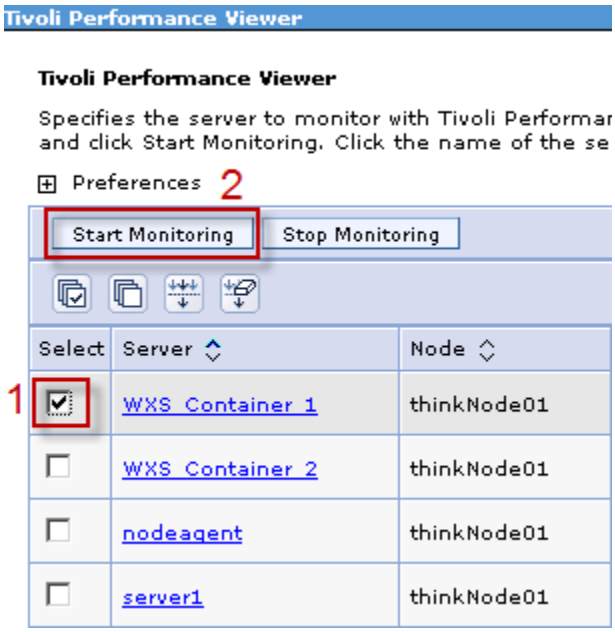


__7. Click on **ObjectGrids** and use the same technique as above to enable the Transaction response time counter.

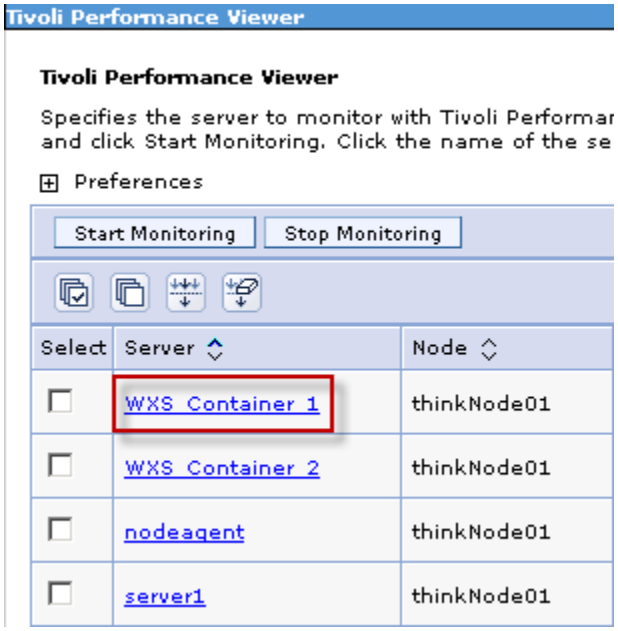


__8. Return to the Administration Console navigation tree. Expand **Monitoring and Tuning → Performance Viewer** and select **Current activity**

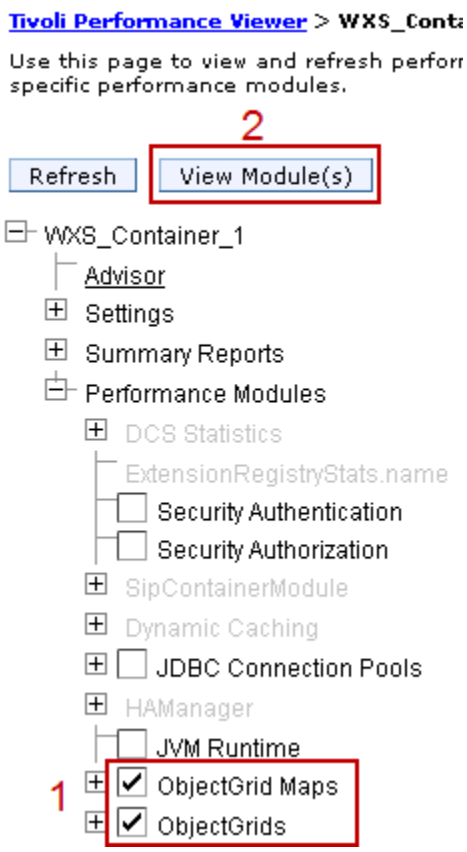
__9. Select **WXS_Container_1** and click the **Start Monitoring** button.



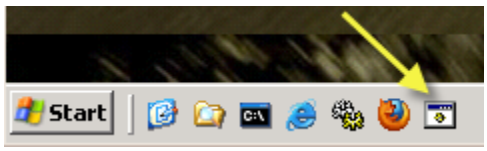
__10. Click on **WXS_Container_1**



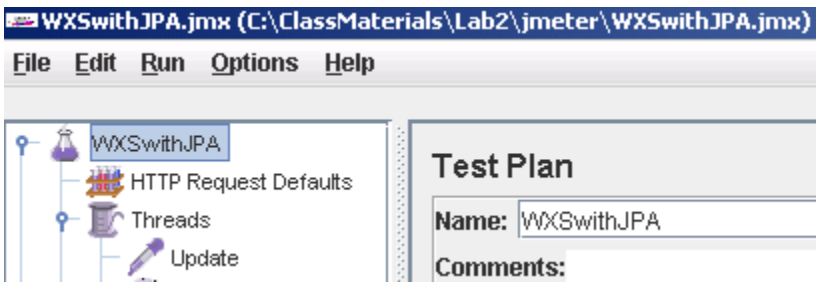
__11. Expand **Performance Modules**. Select **ObjectGrid Maps** and **ObjectGrids**. Click the **View Module(s)** button.



__12. To add additional information into the WebSphere eXtreme Scale grid, you will use Apache JMeter. **Click** on the shortcut link on the Quick Launch toolbar to start JMeter



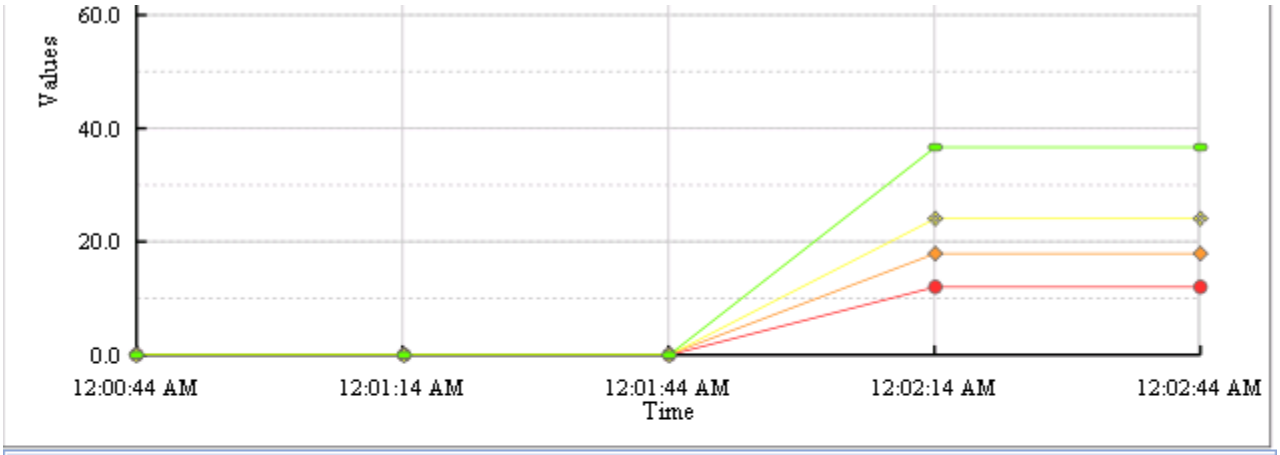
__13. From the menbar, select **File → Open** and choose **`c:\ClassMaterials\Lab2\jmeter\WXSwithJPA.jmx`**



__14. From the JMeter menubar, select **Run → Start**

__15. Return to the WebSphere Administration console. A graph of the WebSphere eXtreme Scale performance data is shown.

It is also helpful to click on the **View Table** button to see the data in table format.



Reset To Zero | Clear Buffer | View Table | Show Legend

Select	Marker	Name	Value	Scale Update	Scaled Value
ObjectGrid Maps					
☑		Map hit rate	12.0	1.0	12.0
☑		Number of map entries	179.0	0.1	17.9
☑		Batch update time for the loader.	24.066668	1.0	24.066668
ObjectGrids					
☑		Transaction response time	36.666668	1.0	36.666668

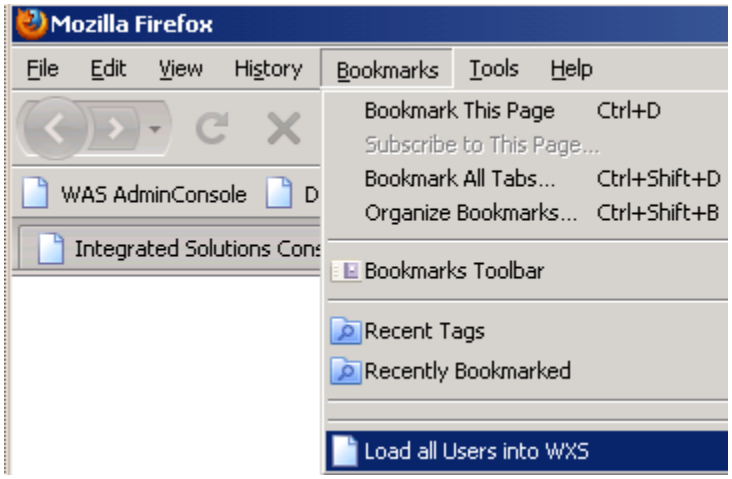
__16. It is also helpful to click on the **View Table** button to see the data in tabular format.

Start Logging

Time	ObjectGrid Maps Map hit rate	ObjectGrid Maps Number of map entries	ObjectGrid Maps Batch update time for the loader.
8:55:02 PM	6.00	188.00	23.17
8:54:32 PM	6.00	188.00	23.17
8:54:02 PM	6.00	188.00	23.17
8:53:32 PM	0.00	2.00	0.00
8:53:01 PM	0.00	2.00	0.00

In this example, the ObjectGrid Map hit rate is not high – only 6%.

__17. To improve the cache hit rate, return to your browser. Enter **CTRL-T** to create a new tab. From the Bookmarks menubar, select **Load all Users into WXS.** This loads all 1000 Users into the data grid. A *Loaded Users* message will appear when this is complete.



__18. Return to JMeter. Select **Run → Start**

__19. From the WAS AdminConsole, you will now see 1000 ObjectGrid Map entries and the ObjectGrid Map hit rate improves significantly.

Start Logging

Time	ObjectGrid Maps Map hit rate	ObjectGrid Maps Number of map entries	ObjectGrid Maps Batch update time for the loader.
9:06:35 PM	42.00	1000.00	15.93
9:06:05 PM	42.00	1000.00	15.93
9:05:35 PM	42.00	1000.00	15.93

Appendix A. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have

been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

Appendix B. Trademarks and copyrights

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

IBM	AIX	CICS	ClearCase	ClearQuest	Cloudscape
Cube Views	DB2	developerWorks	DRDA	IMS	IMS/ESA
Informix	Lotus	Lotus Workflow	MQSeries	OmniFind	
Rational	Redbooks	Red Brick	RequisitePro	System i	
System z	Tivoli	WebSphere	Workplace	System p	

Adobe, Acrobat, Portable Document Format (PDF), and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

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

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

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

ITIL is a registered trademark and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

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

NOTES

NOTES

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM Software