

WebSphere Application Server Version 6.1

Sales and Technical Enablement Workshop

Lab 02 – Application Server Toolkit

Introduction

The *Application Server Toolkit (AST)* provides basic support for the creation of new applications targeting WebSphere Application Server V6.1. This includes wizards and tools for creating new Web applications, Web services, portlets, and EJBs, as well as annotation based programming support, new administration tools for the creation and maintenance of wsadmin Jython files, and tools to edit WebSphere-specific bindings and extensions.

Lab Requirements

This lab assumes that the following setup is complete prior to starting the lab:

- VMware Player 1.0.x or VMware Workstation v5.5.x installed on your machine. A free VMware player is available from <http://www.vmware.com/products/player/>
- A machine with 2 GB is RAM is preferred.

Overview

In this exercise you will explore the Application Server Toolkit by developing a small J2EE application. This lab does not require any prior J2EE development or programming experience.

The application to be developed is named HitCount. HitCount is part of the *Default Application* that WebSphere Application Server v6.1 provides as a default configuration that allows administrators to easily verify that the Application Server is running.

The HitCount application demonstrates how to increment a counter using a variety of methods, including:

- A servlet instance variable
- An HTTP session
- An enterprise bean

You can instruct the servlet to execute any of these methods within a transaction that you can commit or roll back. If the transaction is committed, the counter is incremented. If the transaction is rolled back, the counter is not incremented.

The enterprise bean method uses a container-managed persistence enterprise bean that persists the counter value to a Cloudscape database.

When using the enterprise bean method, you can instruct the servlet to look up the enterprise bean, either in the WebSphere global namespace, or in the namespace local to the application.

The URL for the HitCount application is: `http://localhost:port/HitCount.jsp`.

Part 1: Start the Application Server Toolkit

As an introduction to the Application Server Toolkit, start the Workbench and begin developing the HitCount application.

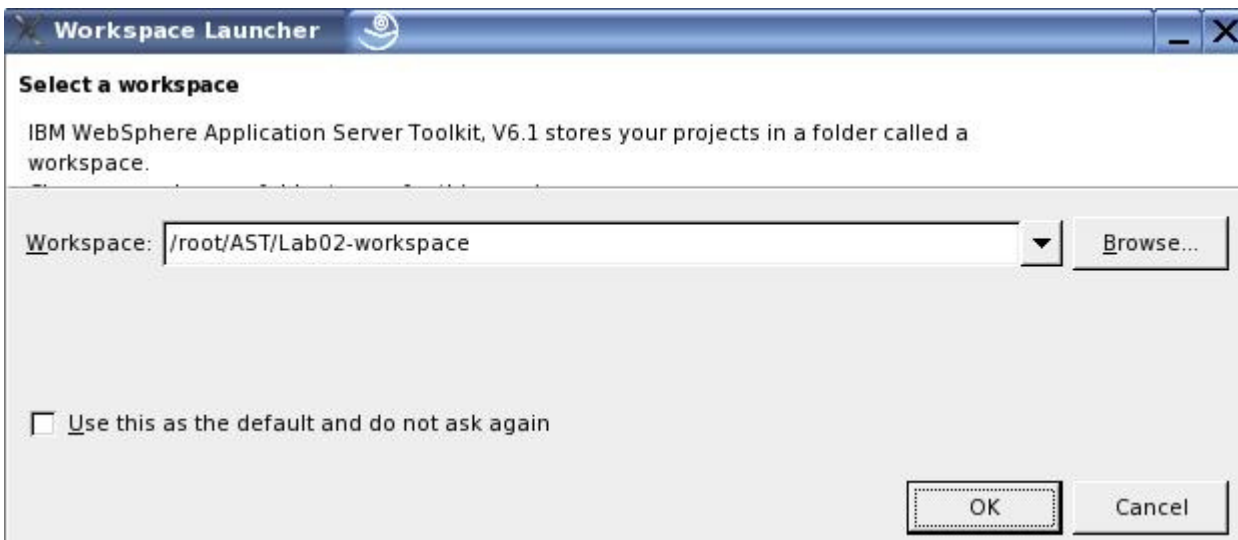
1. From the SLES Desktop, locate the KDE Panel at the bottom of the workspace. Click on the 'N' icon.



2. In the menu, select **IBM WebSphere → Application Server Toolkit V6.1 → Application Server Toolkit V6.1**. This will start the AST. Alternatively, you can start the AST from the command line using `/opt/IBM/AST61/ast`



3. When the Workbench is launched the first thing you see is a dialog that allows you to select where the workspace should be located. The workspace is the directory where your work will be stored. Enter a workspace of `/root/AST/Lab02-workspace` and click **OK** to continue.



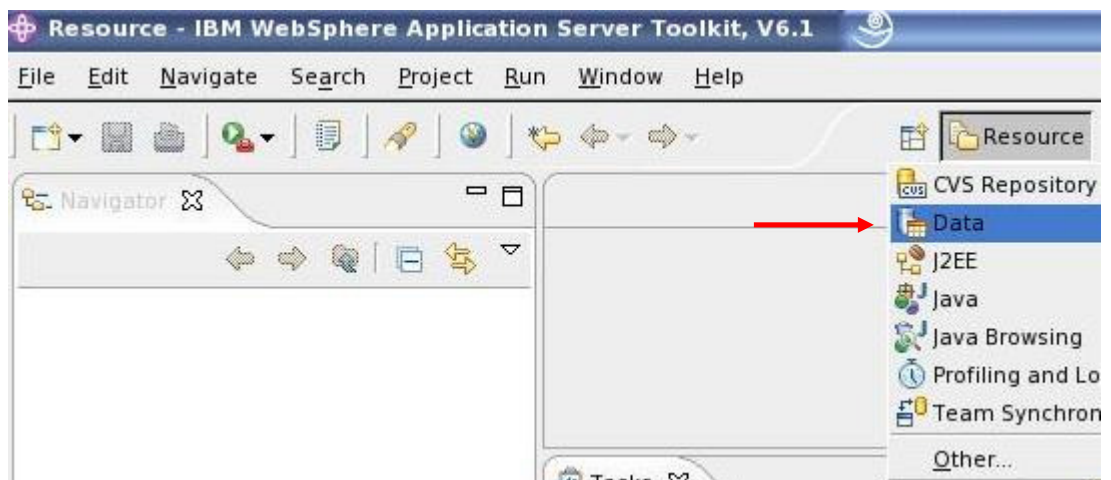
- ____ 4. From the Application Server Toolkit Welcome page, click on **Workbench – Go to the workbench**



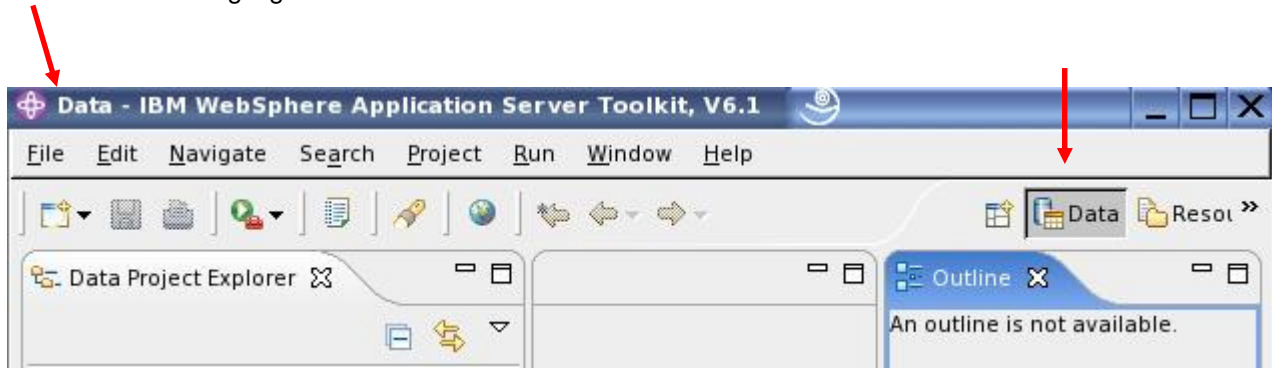
- ____ 5. Initially, in the first Workbench window that is opened, a **Resource** perspective defines the initial set and layout of views in the Workbench window. Within the window, each perspective shares the same set of editors.
- ____ a. A shortcut bar appears in the top right corner of the window that allows the user to open new perspectives and switch between ones already open. From the shortcut bar, click on the **Open Perspective** icon.



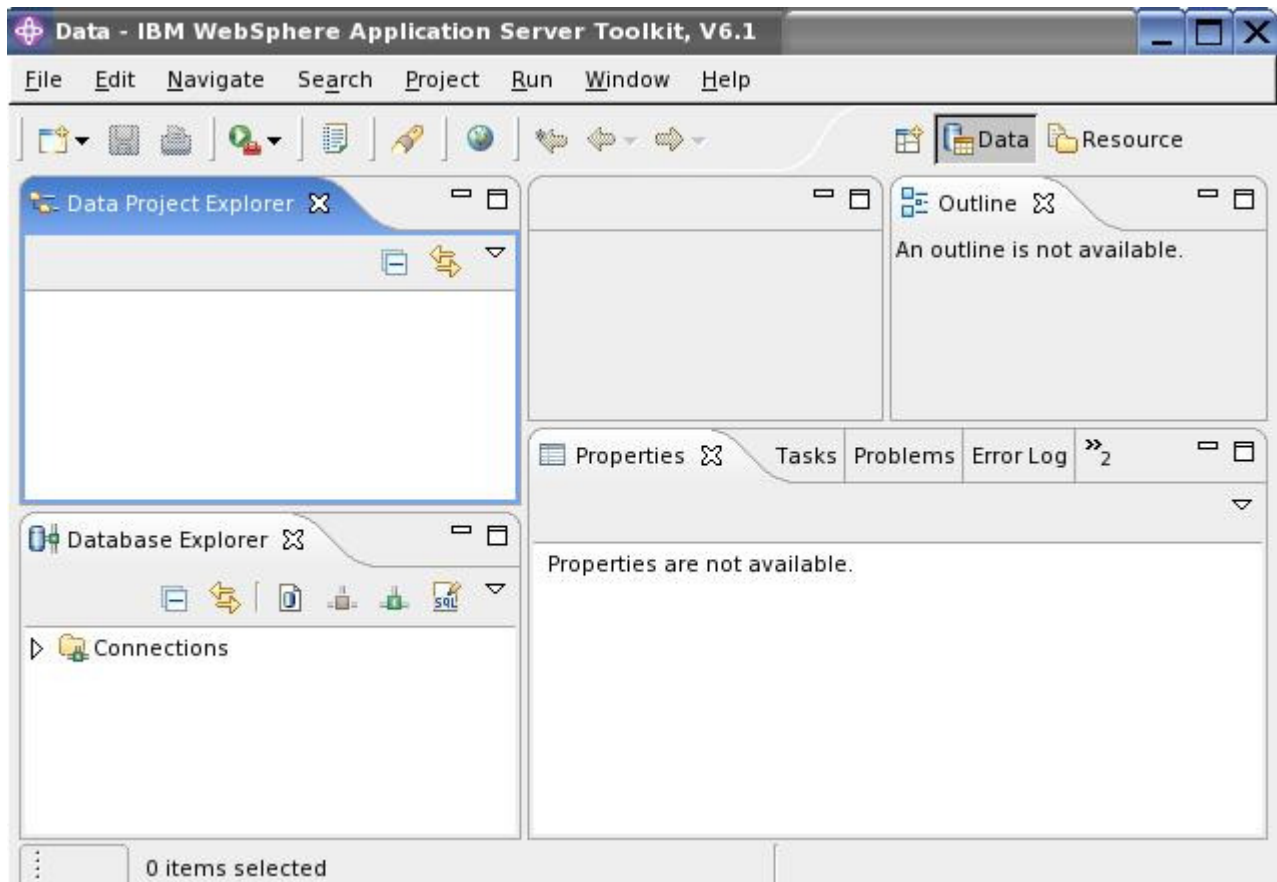
- ____ b. From the pulldown menu, select **Data**



- ___ c. The name of the active perspective is shown in the title of the window and its item in the shortcut bar is highlighted.



- ___ d. In the **Data** perspective, there are different views available, such as **Data Project Explorer**, **Outline**, and **Database Explorer**.



Part 2: Developing data access applications

By using the wizards and views in the relational database tools you can define and work with databases; define and work with tables, views, and filters; create and work with SQL statements; create and work with DB2(R) UDB routines (such as DB2 UDB stored procedures and user-defined functions); and create and work with SQLJ files. You can also generate SQL DDL, and XML files.

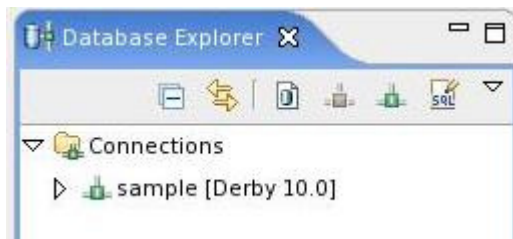
In this section of the lab, you will create an **IBM Cloudscape Version 10.1** database with the schema and tables required for the HitCount application.

The Cloudscape v10.1 code base, which the open source community calls Derby, is a product of the Apache Software Foundation open source relational database project.

WebSphere Application Server supports the **IBM Cloudscape Version 10.1** database. This database provides Derby Version 10.1 binaries, NLS enablement, QA, and IBM problem support. Typical use of the IBM Cloudscape Version 10.1 database is for test and development only.

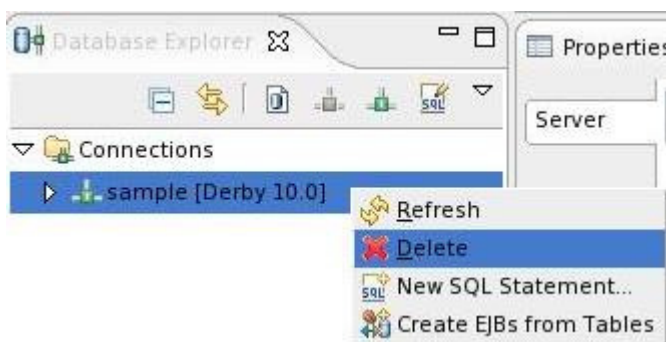
1. Create a new database connection

- a. Locate the **Database Explorer** view. In the Database Explorer view, you can connect to databases and view their designs and objects. Expand **Connections** to reveal the **sample [Derby 10.0]** connection.

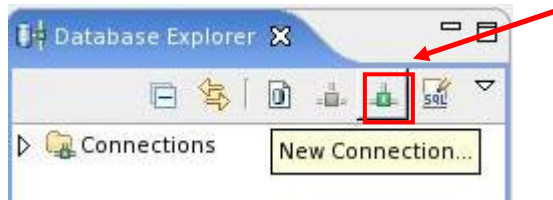


- b. Highlight the **sample [Derby 10.0]** connection, right-click and select **Delete**.

Note: This action deletes the database connection; it does not delete the database.

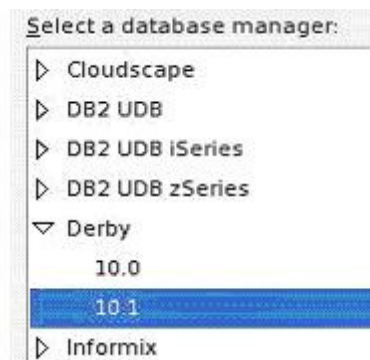


___ c. Create a new database connection by clicking on the **New Connection...** icon.



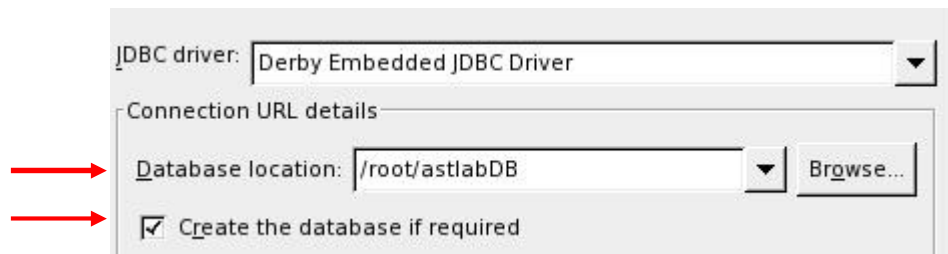
___ d. In the New Connection wizard, under the **Select a database manager** section, choose a database manager of **Derby 10.1**.

When using the Application Server Toolkit and WebSphere Application Server, you will see the term "Derby" rather than "Cloudscape" used in places such as the Database Explorer, Administrative Console, the ejbdeploy command, and others.



___ e. In the **Connection URL details** section, enter a **Database location** of **/root/astlabDB**

The **Create the database if required** option is enabled to create the database on first access.

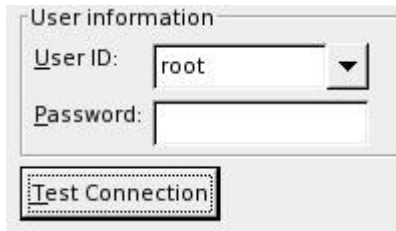


___ f. Set the **Class location** to **/opt/IBM/WAS61/AppServer/derby/lib/derby.jar** by using the **Browse** button



___ g. Locate the **User information** section.

Note: This lab is not using database security, but Cloudscape Version 10.1.x (Derby) can be secured using a User ID and password. Additionally, higher levels of security can be applied to the data store using the features of the underlying database, for example, Cloudscape Version 10.1.x (Derby) allows the whole database to be encrypted.

A dialog box titled "User information" with two input fields: "User ID:" containing the text "root" and a dropdown arrow, and "Password:" which is empty. Below the fields is a button labeled "Test Connection".

___ h. Press the **Test Connection** button. This should return a **Connection to Derby is successful** message. This creates the database in the location `/root/astlabDB`.



___ i. Click **OK** to close the Test Connection dialog

___ j. Click **Finish** to close the New Connection wizard.

___ k. In the **Database Explorer** view, expand Connections to show the **astlabDB [Derby 10.1]** connection and the newly created database.



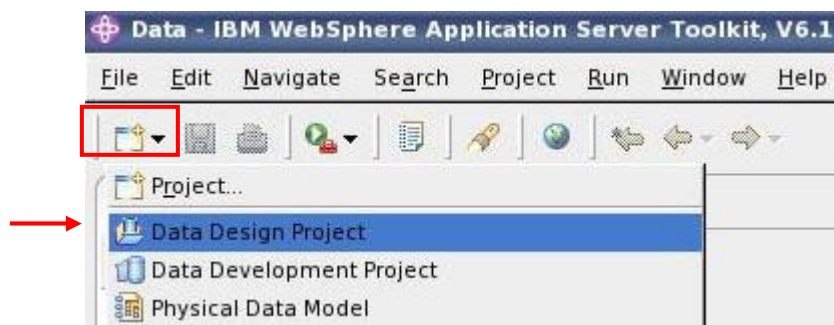
2. Create a Data Design Project

In the previous section, you created a new **astlabDB** connection and database. You will now add tables and columns to this database for the HitCount application. As a preview, the data definition for the table and columns to be created is shown below:

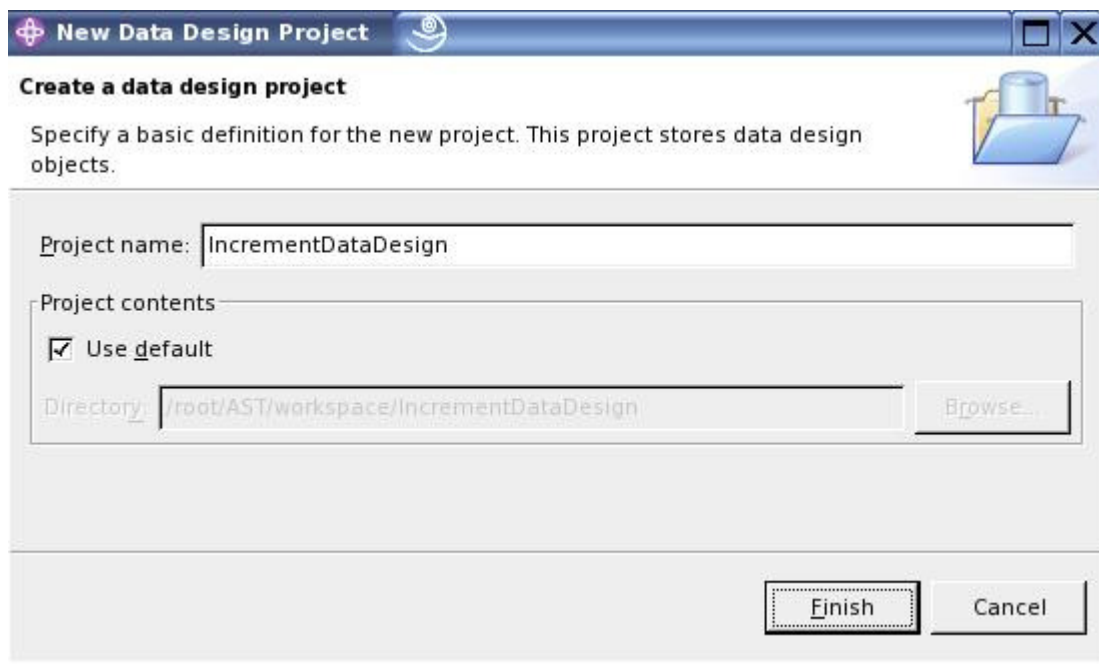
```
CREATE TABLE INCREMENT (  
    PRIMARYKEY VARCHAR(250) NOT NULL,  
    THEVALUE INTEGER NOT NULL  
);
```

```
ALTER TABLE INCREMENT ADD CONSTRAINT PK_INCREMENT PRIMARY KEY (PRIMARYKEY);
```

- a. From the main toolbar, click on the **New Project** icon and then select **Data Design Project**



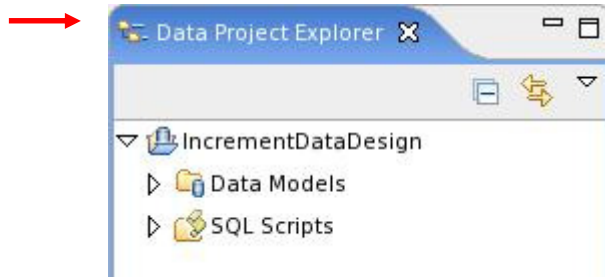
- b. Enter a project name of **IncrementDataDesign** and click **Finish**



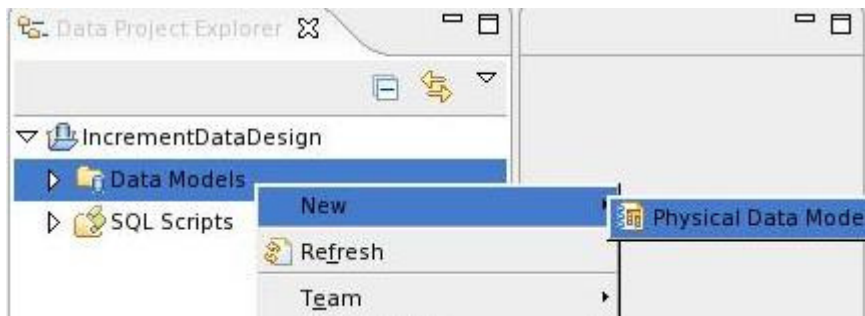
- ___ c. Locate the **Data Project Explorer** view. In the Data Project Explorer, you work locally with data objects.

The Data Project Explorer can hold:

- Data design projects, which are used for database design and information integration. Includes physical data models, logical data models, domain models, glossary models, XSD models, and scripts.
- Data development projects, which include DB2 Universal Database™ stored procedures and user-defined functions, Cloudscape and Derby stored procedures, and SQL statements.



- ___ d. Under the IncrementDataDesign project, select **Data Models**. Right-click and select **New** → **Physical Data Model**

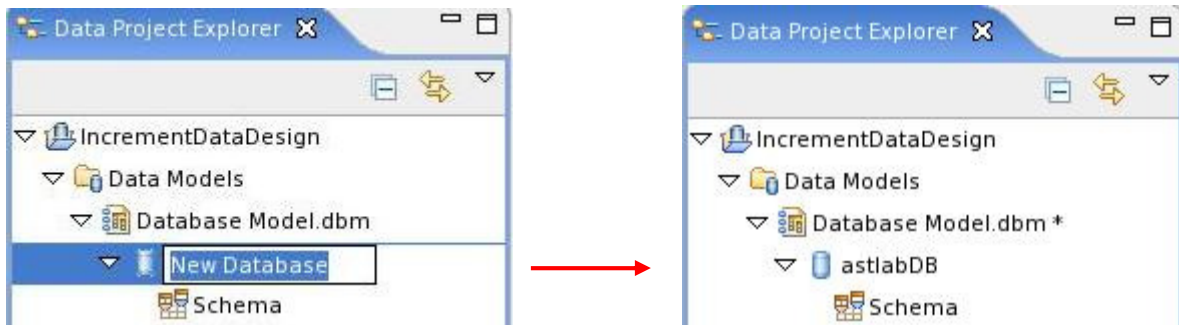


- ___ e. In the Model File panel, specify a database of **Derby**, **Version 10.1** and click **Finish**

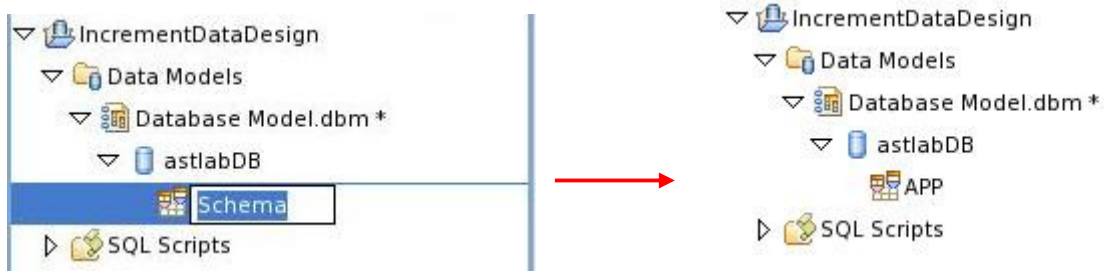
A screenshot of the 'New Physical Data Model' dialog box. The 'Model File' section is visible. It contains the following fields:

- 'Destination folder': /IncrementDataDesign
- 'File name': Database Model
- 'Database': Derby (selected from a dropdown)
- 'Version': 10.1 (selected from a dropdown)
- 'Create from template': selected (radio button)

- ___ f. From the **Data Project Explorer** view, notice under the Database Model there is a database with a name of **New Database**. Select **New Database** and then press the **F2** key. Change the database name to **astlabDB**.



- ___ g. The default database schema name is **Schema**. Change the schema name to **APP** using the **F2** key, similar to the previous step.

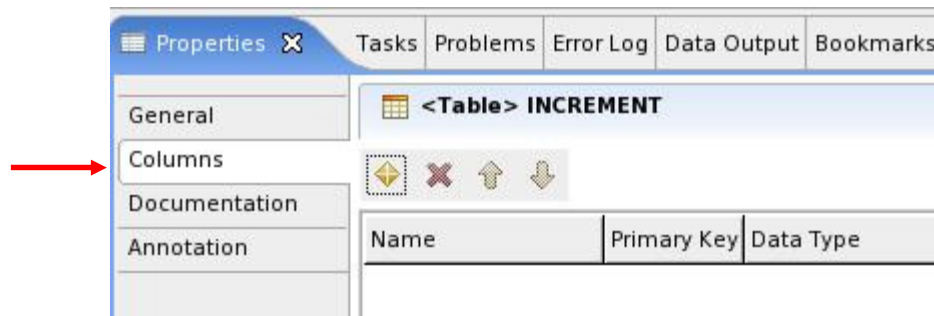


- ___ h. Select the **APP** schema. Right-click and select **Add Data Object → Table**



- ___ i. The default table name is **Table1**. Change the table name to **INCREMENT**. Use the **F2** key if necessary.

- ___ j. Select the **INCREMENT** table in the **Data Project Explorer** view. Then locate the **Properties** view (look in the lower portion of the perspective). In the **Properties** view, from the left-side menu, click on the **Columns** tab.

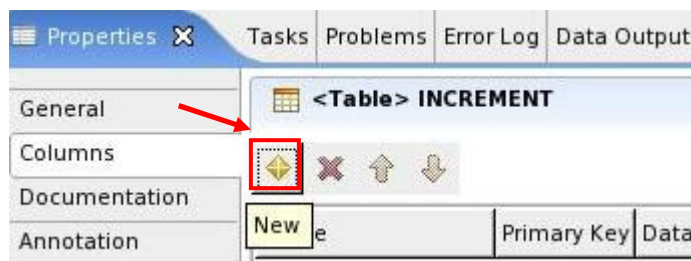


- ___ k. The HitCount application uses a data description shown below. You'll now create these columns for the **INCREMENT** table.

```
CREATE TABLE INCREMENT (
    PRIMARYKEY VARCHAR(250) NOT NULL,
    THEVALUE INTEGER NOT NULL
);
```

```
ALTER TABLE INCREMENT ADD CONSTRAINT PK_INCREMENT PRIMARY KEY (PRIMARYKEY);
```

- ___ l. In the **<Table> INCREMENT** workspace, click on the **New** icon to add a column.



- ___ m. Use the data description shown above to enter the values for the **PRIMARYKEY** column. Verify your entry with the picture below.

<Table> INCREMENT					
Name	Primary Key	Data Type	Length	Scale	Not Null
PRIMARYKEY	<input checked="" type="checkbox"/>	VARCHAR	250		<input checked="" type="checkbox"/>

- ___ n. Click the **New** icon to add another column. Use the data description above to enter the values for the **THEVALUE** column.

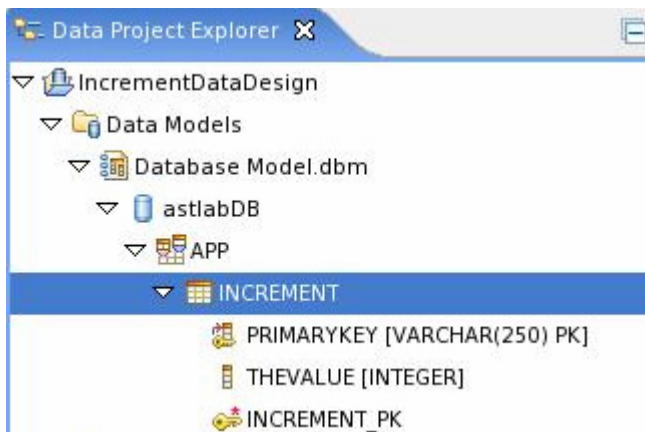
___ o. When finished, the columns should look as shown below:

<Table> INCREMENT					
Name	Primary Key	Data Type	Length	Scale	Not Null
PRIMARYKEY	<input checked="" type="checkbox"/>	VARCHAR	250		<input checked="" type="checkbox"/>
THEVALUE	<input type="checkbox"/>	INTEGER			<input checked="" type="checkbox"/>

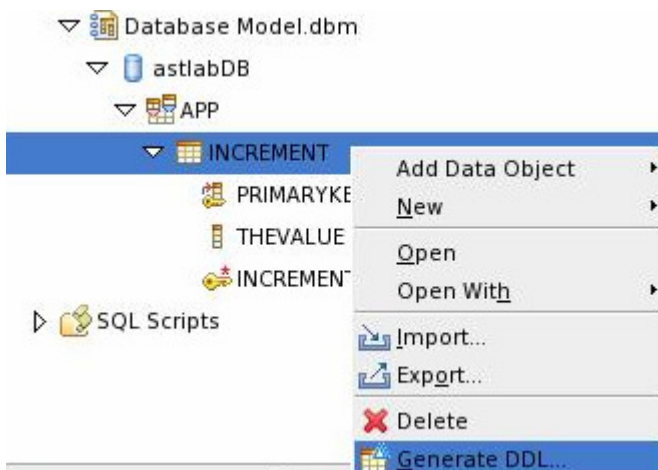
___ p. Use **CTRL-S** to save your changes.

___ 3. In the previous steps, you created the table and columns definitions for the **INCREMENT** table in a Database Model. Now use the Database Model to create these items in the physical **astlabDB** database.

___ a. In the **Data Project Explorer** view, expand the **INCREMENT** table. The columns and types are displayed.



___ b. Select **INCREMENT**. Right-click and select **Generate DDL...**

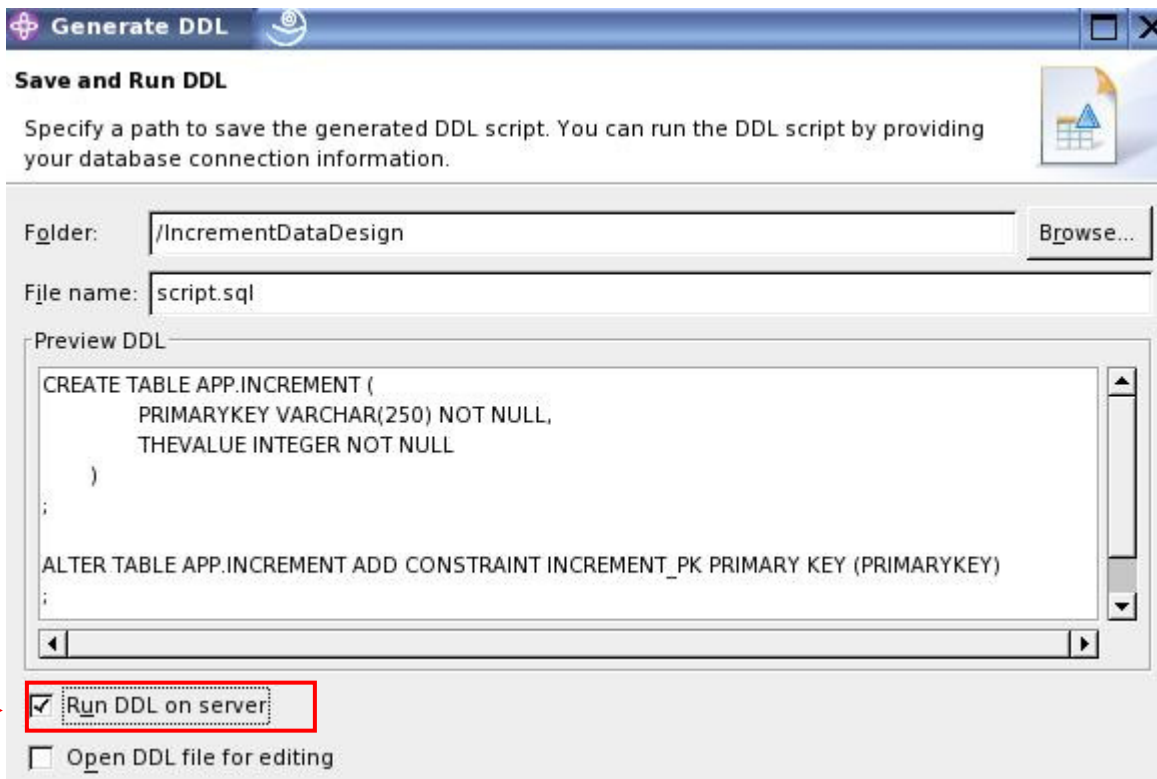


__ c. In the Options panel, select **Fully qualified name** and click **Next**.



__ d. Click **Next** on the Objects panel

__ e. In the **Save and Run DDL** panel, select the **Run DDL on server** checkbox. Click **Next**



- ___ f. In the Select Connection panel, choose **Use an existing connection**. The existing connection will be the **astlabDB** connection. Click **Next**

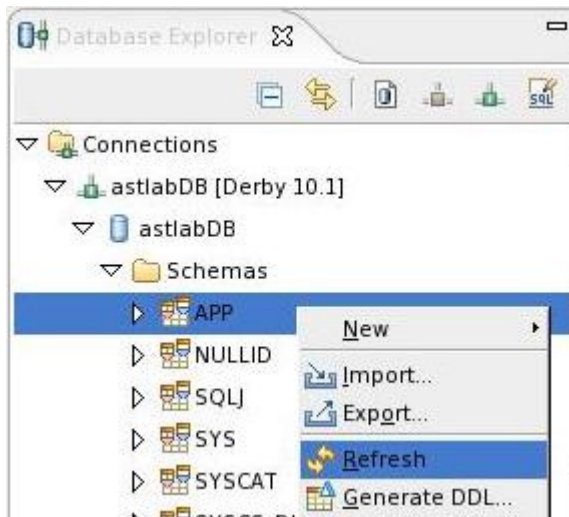


- ___ g. In the Summary panel, click **Finish**
- ___ h. Locate the **Data Output** view. In the Data Output view, you can see the messages, parameters, and results that are related to the database objects that you work with. In this example, the INCREMENT table and columns will be added to the database. The **Status** should be **Success** for the two DDL Generation statements. In the **Messages** tab, you can see the DDL that was run.

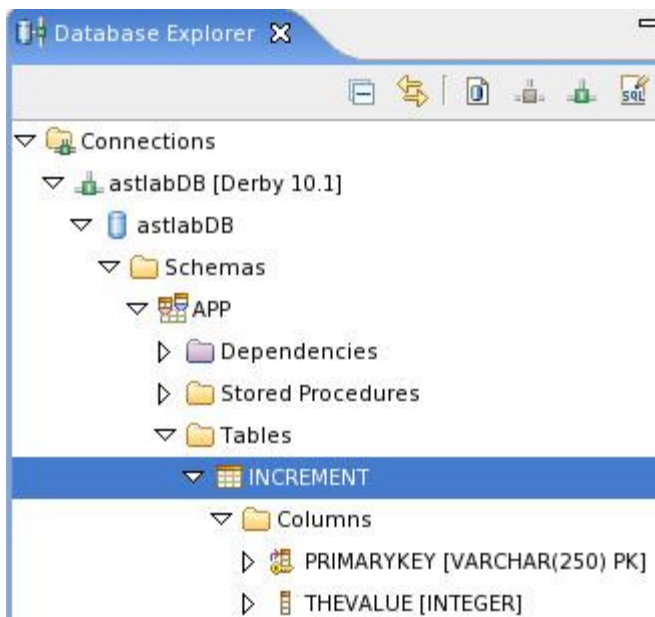
Properties	Tasks	Problems	Error Log	Data Output	Bookmarks
Status	Action	Object Name	DDL Generation		
✓ Success	Run	DDL Generation	Messages	Parameters	Results
✓ Success	Run	DDL Generation	Starting run		
			ALTER TABLE APP.INCREMENT ADD C KEY (PRIMARYKEY)		
			Run successful		

4. Create and examine data in the **INCREMENT** table

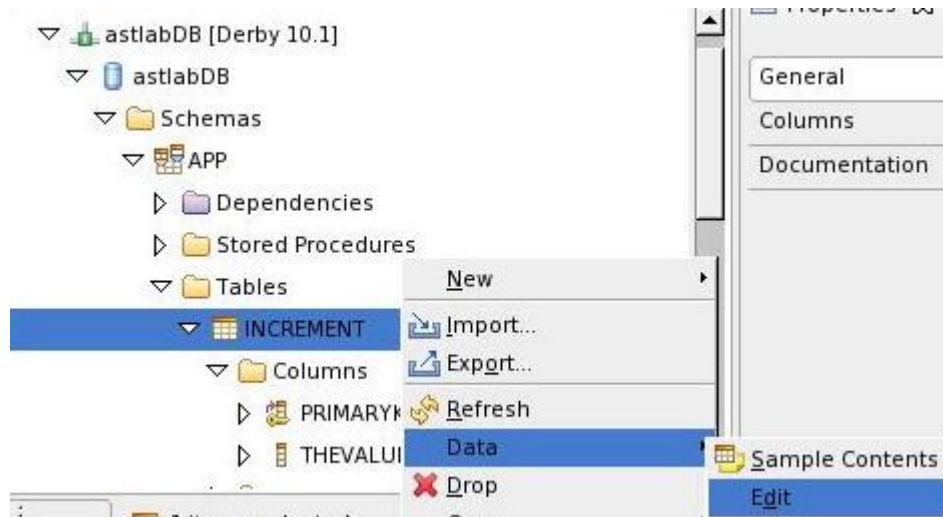
- a. In the **Database Explorer** view (not the **Data Project Explorer** view), expand the view and locate the **APP** schema. Right-click and select **Refresh**



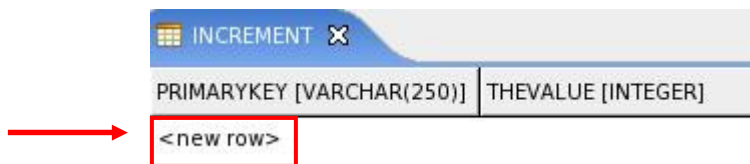
- b. Expand the **APP** schema, expand **Tables**, and select the **INCREMENT** table



___ c. Right-click on the **INCREMENT** table and choose **Data → Edit**



___ d. Locate the **INCREMENT** table view. Click on **<new row>**

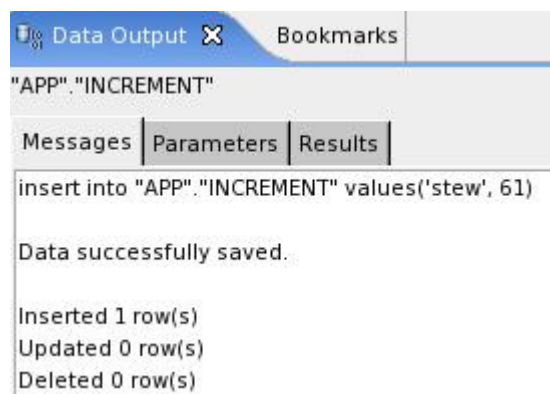


___ e. Enter values of **stew** and **61** for the PRIMARYKEY and THEVALUE columns, respectively.



___ f. Press **CTRL-S** to save.

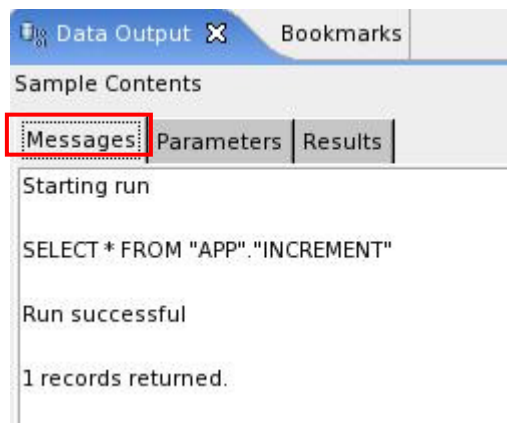
___ g. Locate the **Data Output** view. Confirm the **INSERT** into the **astlabDB** database was successful.



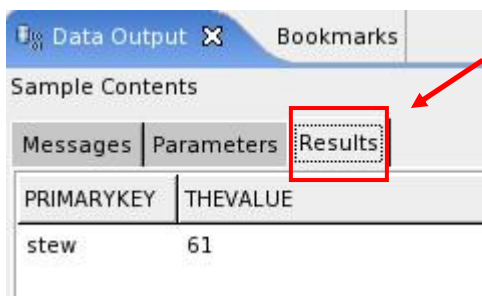
- ___ h. You can continue to explore the **INCREMENT** table view by making additional INSERT, DELETE, and UPDATE changes of your own choosing.
- ___ i. Returning to the Database Explorer view, the **INCREMENT** table should still be selected. Right-click and select **Data → Sample Contents**



- ___ j. From the **Data Output** view, in the **Messages** section, examine the SQL SELECT statement that ran.



- ___ k. Click on the **Results** tab to show the output from the SELECT statement



- ___ l. Close the **INCREMENT** view when you are finished.
- ___ m. Close the **Database Model.dbm** view

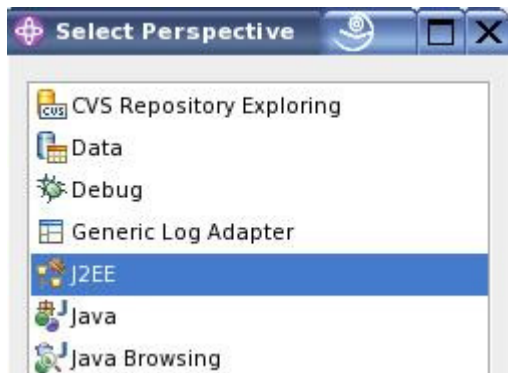
Part 3: Creating an enterprise application

You can develop, assemble and deploy enterprise applications using the Application Server Toolkit. Enterprise applications are applications that conform to the J2EE specification, and are packaged in enterprise archive (EAR) files. In this section of the lab, you will create an Enterprise Application for the HitCount application.

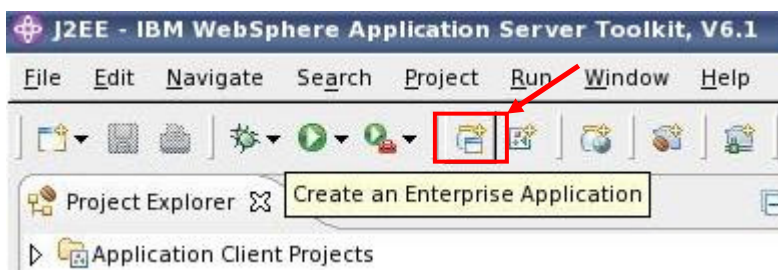
- ___ 1. Switch to a **J2EE** perspective
 - ___ a. Click on the **Open Perspective** icon



- ___ b. Select **Other**
 - ___ c. Select **J2EE** and click **OK**



- ___ 2. From the main toolbar, click on the **Create an Enterprise Application** icon



- ___ a. Enter a Project Name of **HitCount**.

The **Target runtime** option affects the compilation and runtime settings by modifying the class path entries for the project. The target runtime of **WebSphere Application Server v6.1 stub** allows development and compilation against the API's of WebSphere Application Server Version 6.1, without requiring a complete WAS runtime environment installed.



New EAR Application Project

EAR Application Project

Create a EAR application.

Project Name:

Project contents

☒ Use default

Directory:

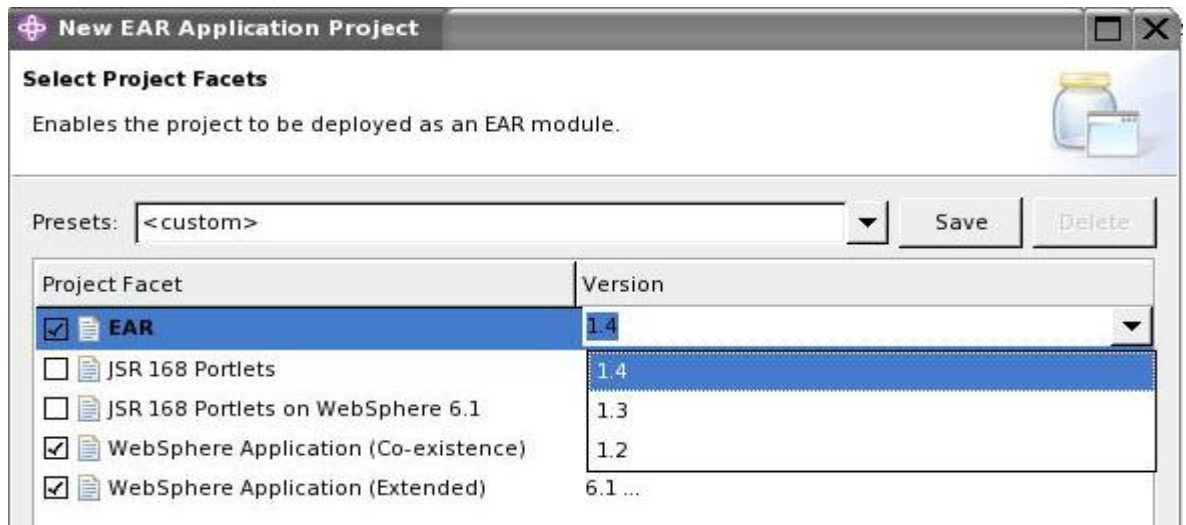
Target runtime:

__ b. Click **Next**

__ c. In the **Select Project Facets** panel, take the defaults selections.

Enterprise application projects can contain one or more project facets, which represent units of functionality in the project. For example, the **JSR 168 Portlets** facet allows for portlet development.

Notice that under the **Version** column, any Version number with three trailing periods (dots) can be selected. For example, the EAR project facet has a default of 1.4. Clicking on **1.4...** shows the entire range of EAR versions.



Select Project Facets

Enables the project to be deployed as an EAR module.

Presets:

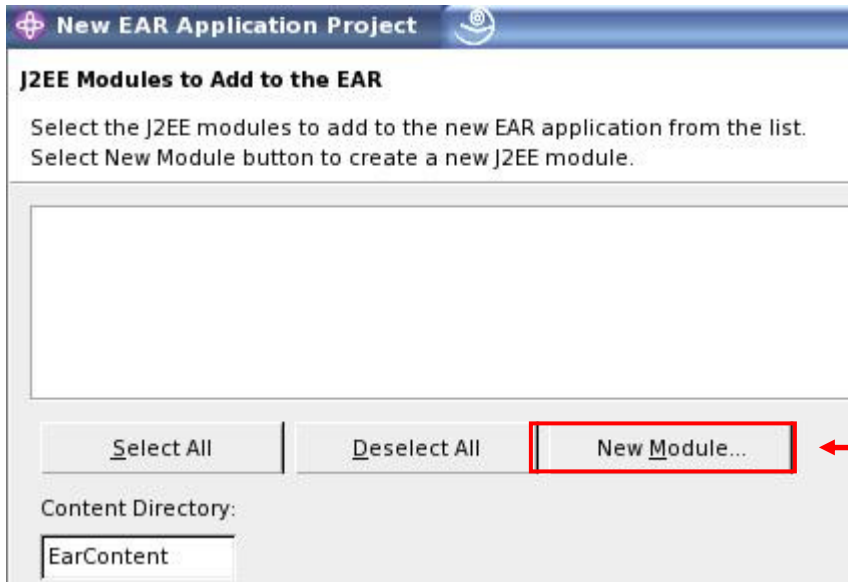
Project Facet	Version
<input checked="" type="checkbox"/> EAR	1.4
<input type="checkbox"/> JSR 168 Portlets	1.4
<input type="checkbox"/> JSR 168 Portlets on WebSphere 6.1	1.3
<input checked="" type="checkbox"/> WebSphere Application (Co-existence)	1.2
<input checked="" type="checkbox"/> WebSphere Application (Extended)	6.1 ...

Two types of **WebSphere Application** project facets appear on the Select Project Facets page:

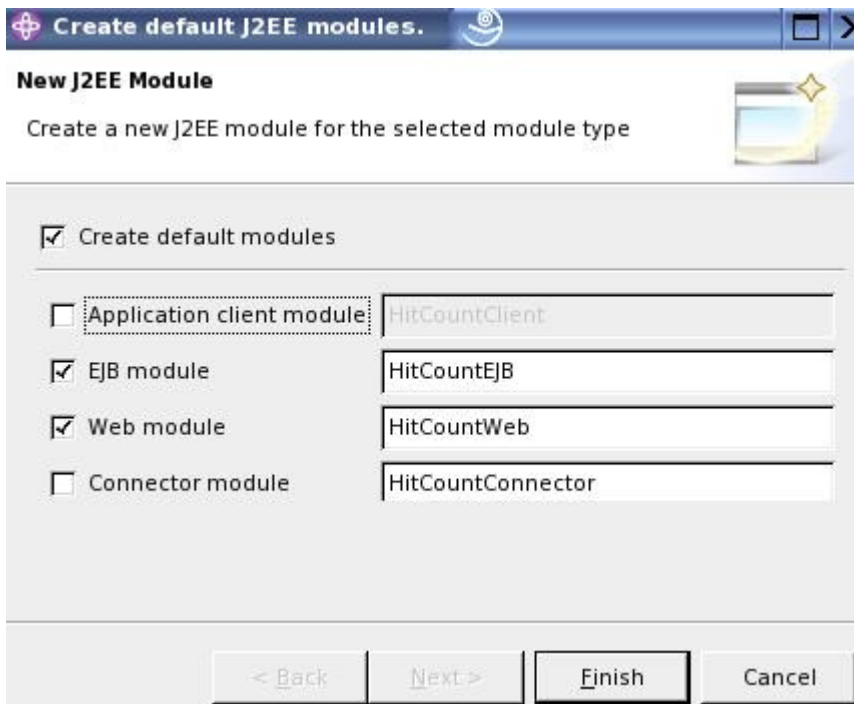
1. **Co-existent facet** -- applies to applications that deploy to other servers.

2. **Extended facet** -- applies to applications that use WebSphere-specific functionality that is not covered in the J2EE standards and therefore cannot be deployed to other servers.

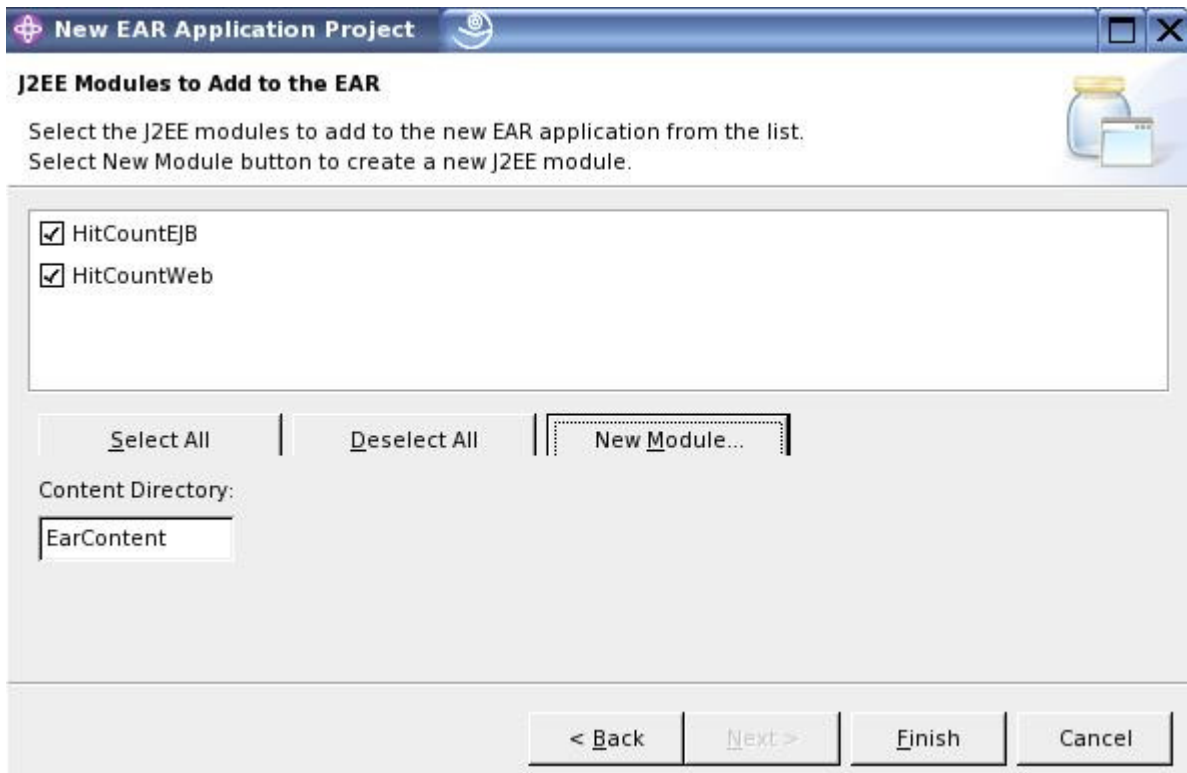
- ___ d. Click **Next** when you are done investigating this panel
- ___ e. In the 'J2EE Modules to Add to the EAR' panel, click the **New Module** button



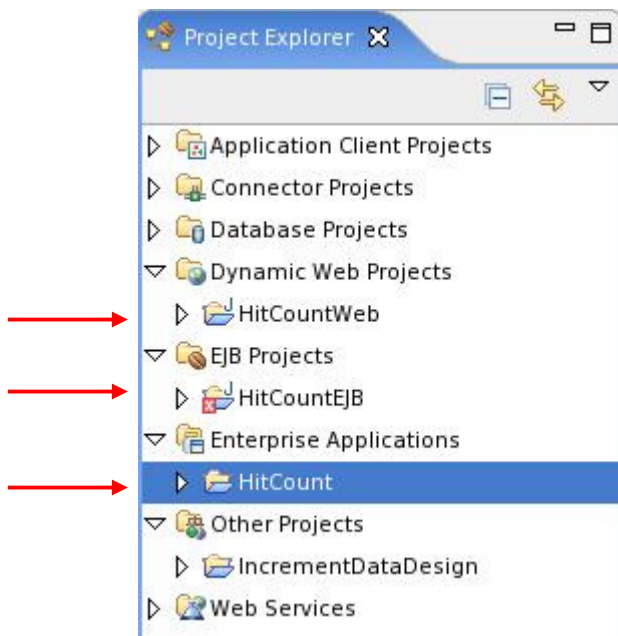
- ___ f. In the New J2EE Module panel, **uncheck** Application Client module and Connector module. This leaves an EJB module and a Web module selected. Click **Finish** and wait for a short period while the workspace builds.



- ___ g. You'll be returned to the **J2EE Modules to Add to the EAR** panel. Click **Finish**. Wait a short period for this task to complete.



- ___ h. Returning to the **Project Explorer** view, you can now see the results of the Enterprise Application creation wizard. Expanding Dynamic Web Projects, EJB Projects, and Enterprise Applications will show the new HitCount projects. **Ignore** any problems in the HitCountEJB project for now. You will address this in the next section.



Part 4: Creating Enterprise Java Beans

The workbench provides a specialized environment that you can use to develop and test enterprise beans that conform to the distributed component architecture defined in the Sun Microsystems Enterprise JavaBeans™ (EJB) specification. The Application Server Toolkit supports the Enterprise JavaBeans 1.1, 2.0, and 2.1 specification levels.

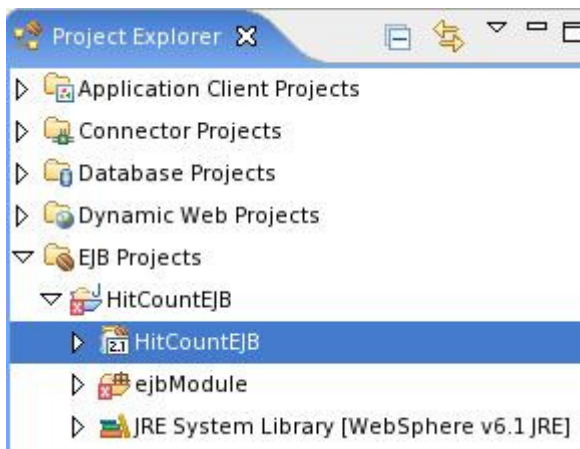
- ___ 1. Create an Increment Enterprise Java Bean using a bottom-up mapping from the **astlabDB** database.

Mapping enterprise bean JAR files (EJB modules) to relational database (RDB) tables enables the EJB modules to access database resources. There are three general approaches for creating your initial mappings between enterprise beans and relational database tables:

- Top-down mapping
- Bottom-up mapping
- Meet-in-the-middle mapping

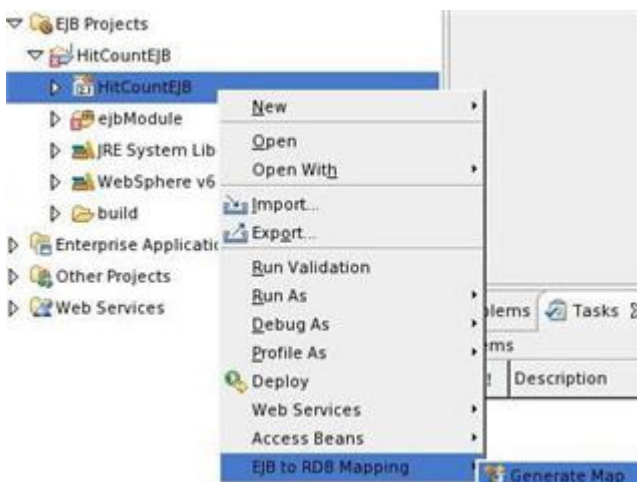
In this lab, you will use the **bottom-up** mapping.

- ___ a. In the **Project Explorer** view, expand the **EJB Projects** section. Ignore the errors for now.

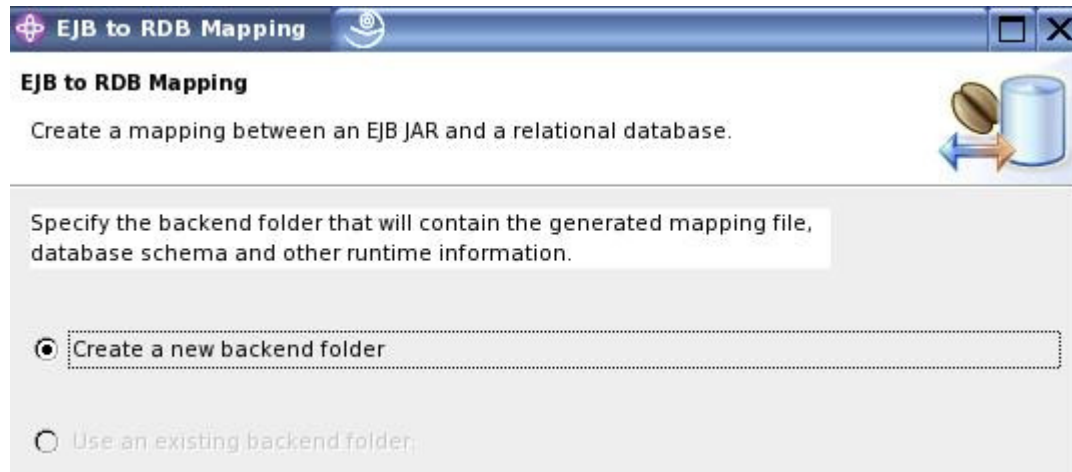


- ___ b. Highlight the **HitCountEJB** deployment descriptor (look at the picture above to verify you are selecting the correct entry)

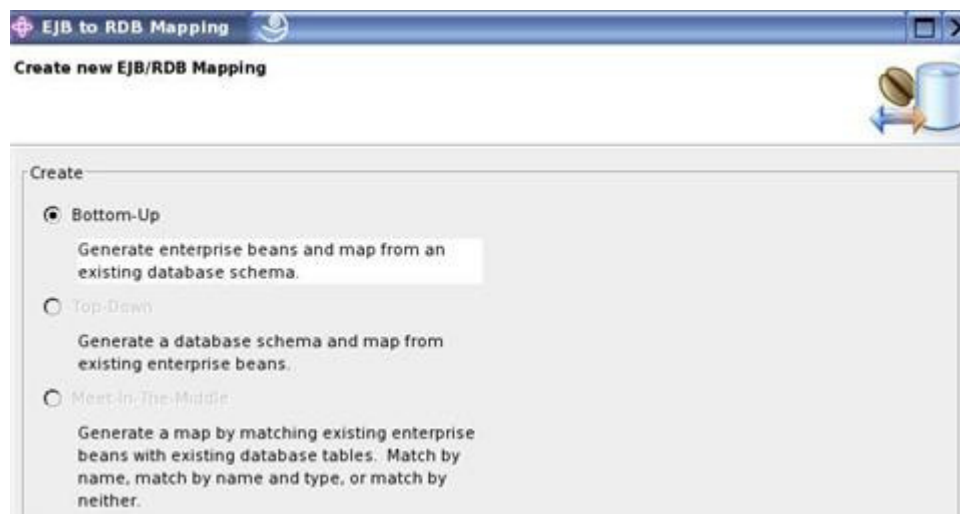
- ___ c. Right-click and select **EJB to RDB Mapping → Generate Map**



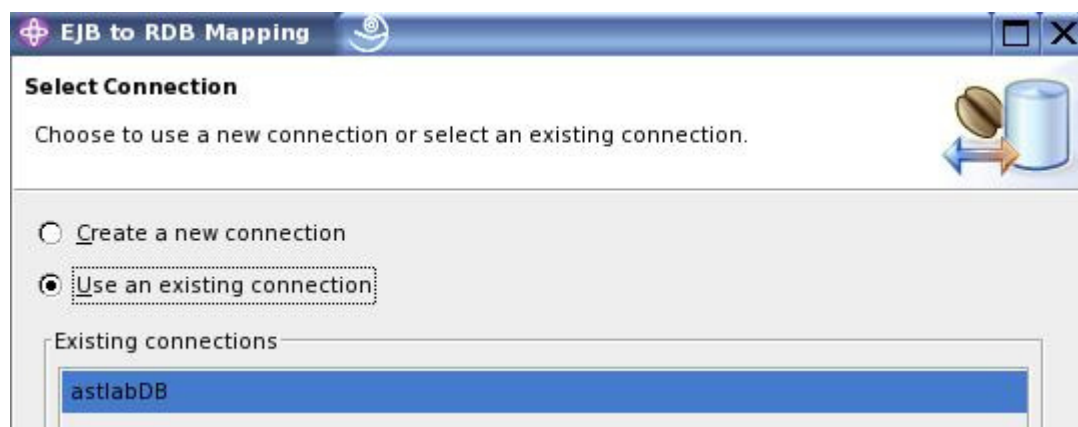
___ d. In the EJB to RDB mapping panel, select **Create a new backend folder** and click **Next**



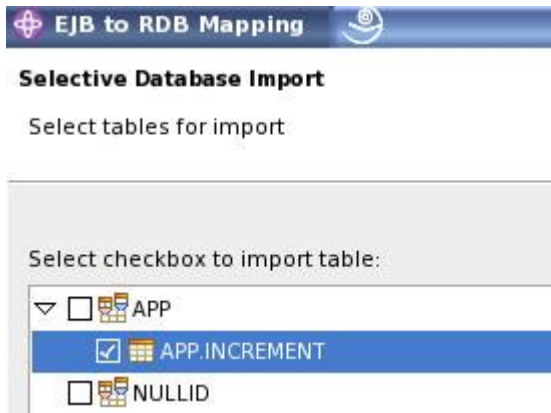
___ e. In the Create new EJB/RDB Mapping panel, select **Bottom-Up** and click **Next**



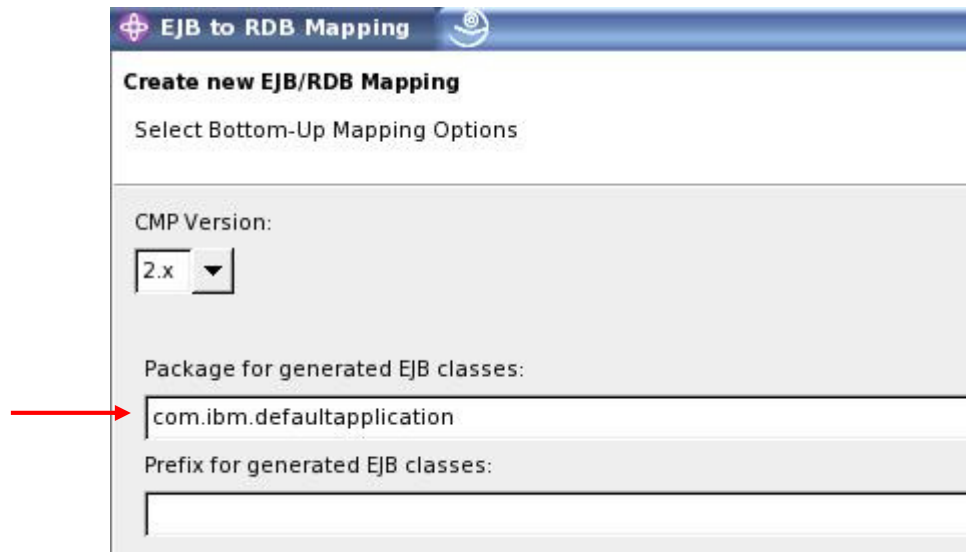
___ f. In the Select Connection panel, select **Use an existing connection** with the existing **astlabDB** connection selected. Click **Next**



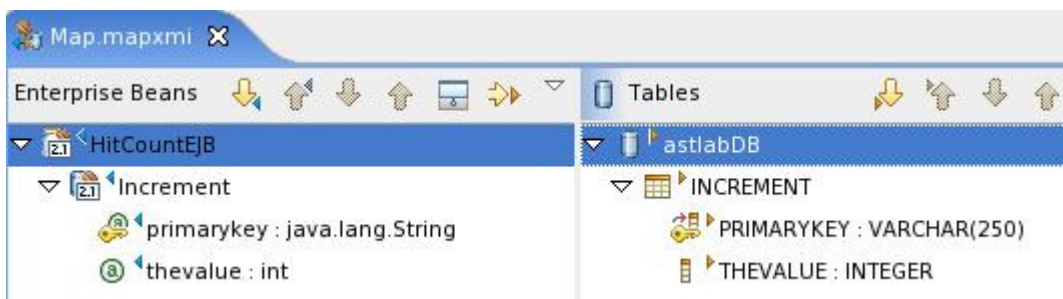
___ g. In the Selective Database Import panel, select the **APP.INCREMENT** checkbox and click **Next**



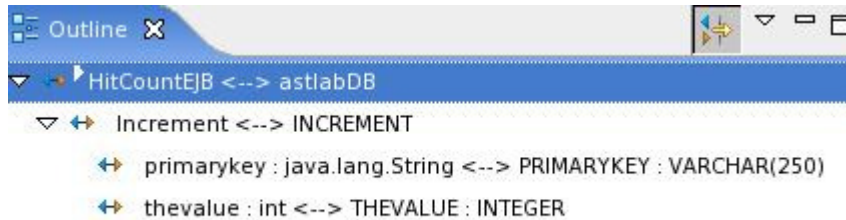
___ h. In the Create new EJB/RDB mapping panel, locate **Package for generated EJB classes** and enter **com.ibm.defaultapplication** Click **Finish** and wait for this task to complete.



___ i. When the mapping is complete, the **Map.mapxmi** view is opened. There is an Increment CMP EJB mapped to the astlabDB database and INCREMENT table.



___ j. Examine the **Outline view** for another look at this mapping.

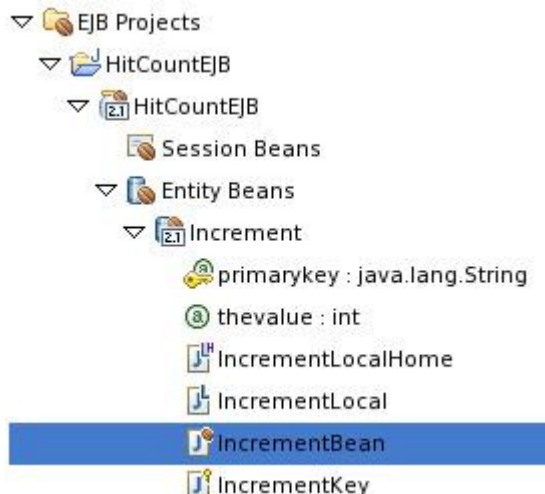


___ k. Close the **Map.mapxmi** view.

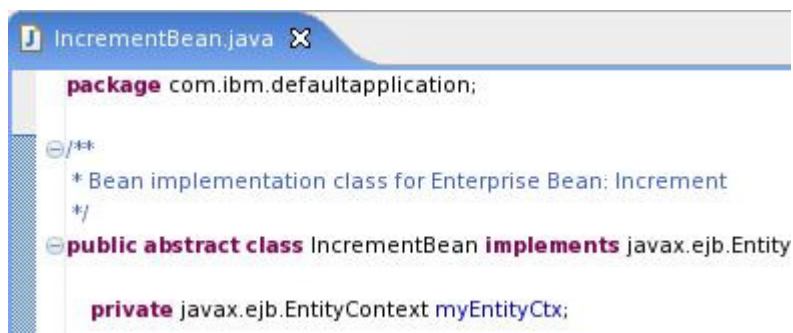
2. Add the application logic to the Increment Bean

You have just created the Increment EJB from a bottom-up database mapping. Now add the application logic to the Increment Bean.

___ a. Returning to the Project Explorer view, expand **HitCountEJB → Entity Beans → Increment** to show the files of the Increment Entity Bean. Select **IncrementBean**



___ b. Double-click on **IncrementBean** to open it in the **Java Editor**



___ c. **Note** – If the font is too small in the **IncrementBean.java** editor:

- 1) From the AST menubar, select **Windows → Preferences...**
- 2) On the left panel, expand **General → Appearance**
- 3) Click on **Color and Fonts**
- 4) In the Workspace, expand **Basic**
- 5) Scroll down and select **Text Font**
- 6) Click on **Use System Font** or **Change...** to set the font
- 7) Click **OK** to accept the changes and close the Preferences window

___ d. The Increment Bean has the following application logic -- when it is called, a counter is incremented. You will now add the **increment()** method to the Increment Bean by copy and paste.

- 1) From the SLES Desktop, locate the KDE Panel at the bottom of the workspace. Click on the **Personal Files** icon.



- 2) Navigate to **/root/WAS61STEW/hands-on/Lab02-AST**. You should see three files.



- 3) Single click on **IncrementCode.txt** to display its contents
- 4) Use your mouse to select entire contents of the file (you can also use **CTRL-A**). Then use **CTRL-C** to copy the entire file contents to the clipboard.

```
// Increment Bean Methods
public int increment() {

    int value = getThevalue();
    value++;
    setThevalue(value);

    return value;
}
```


- 5) Return to the Application Server Toolkit. In the **IncrementBean.java** editor, position your cursor as shown in the picture below. Then use **CTRL-V** (paste) to add in the code.



The screenshot shows the **IncrementBean.java** editor with the following code:

```
package com.ibm.defaultapplication;

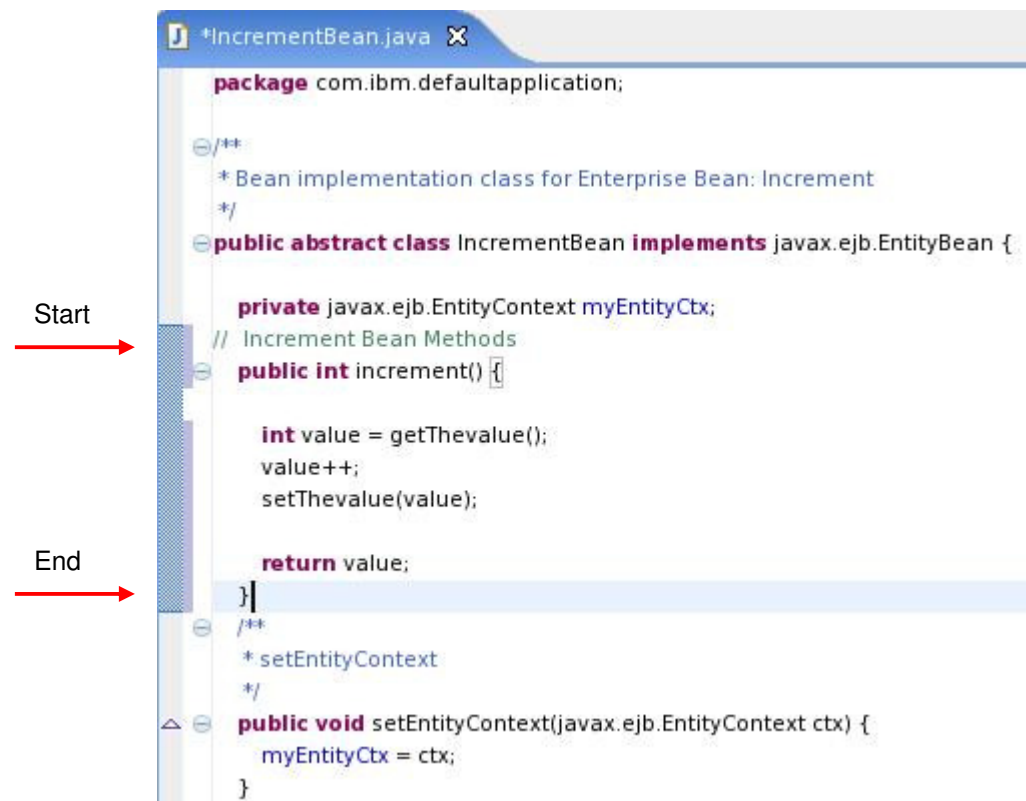
/**
 * Bean implementation class for Enterprise Bean: Increment
 */
public abstract class IncrementBean implements javax.ejb.EntityBean {

    private javax.ejb.EntityContext myEntityCtx;

    /**
     * setEntityContext
     */
    public void setEntityContext(javax.ejb.EntityContext ctx) {
        myEntityCtx = ctx;
    }
}
```

A red arrow labeled "Insert Here" points to the line after the `private javax.ejb.EntityContext myEntityCtx;` declaration.

- 6) After the paste, the **IncrementBean.java** file should appear as shown below.



The screenshot shows the **IncrementBean.java** editor with the following code:

```
package com.ibm.defaultapplication;

/**
 * Bean implementation class for Enterprise Bean: Increment
 */
public abstract class IncrementBean implements javax.ejb.EntityBean {

    private javax.ejb.EntityContext myEntityCtx;

    // Increment Bean Methods
    public int increment() {

        int value = getThevalue();
        value++;
        setThevalue(value);

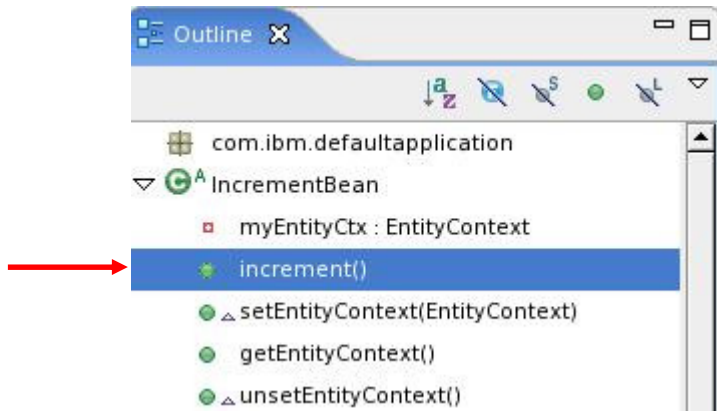
        return value;
    }

    /**
     * setEntityContext
     */
    public void setEntityContext(javax.ejb.EntityContext ctx) {
        myEntityCtx = ctx;
    }
}
```

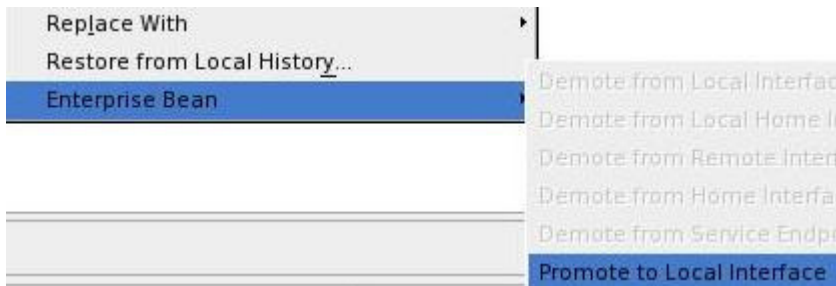
A red arrow labeled "Start" points to the line `// Increment Bean Methods`. Another red arrow labeled "End" points to the closing brace of the `increment()` method.

- 7) Use **CTRL-S** to save the file.

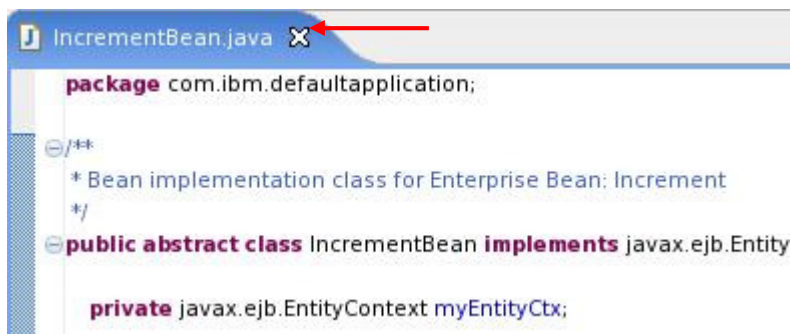
___ e. In the **Outline** view, you will now see the **increment()** method



___ f. Select the **increment()** method, right-click and choose **Enterprise Bean → Promote to Local Interface**. This allows the application logic to be called from clients.



___ g. Close the **IncrementBean.java** view



__ h. Returning to the Project Explorer view, select the **HitCountEJB** deployment descriptor

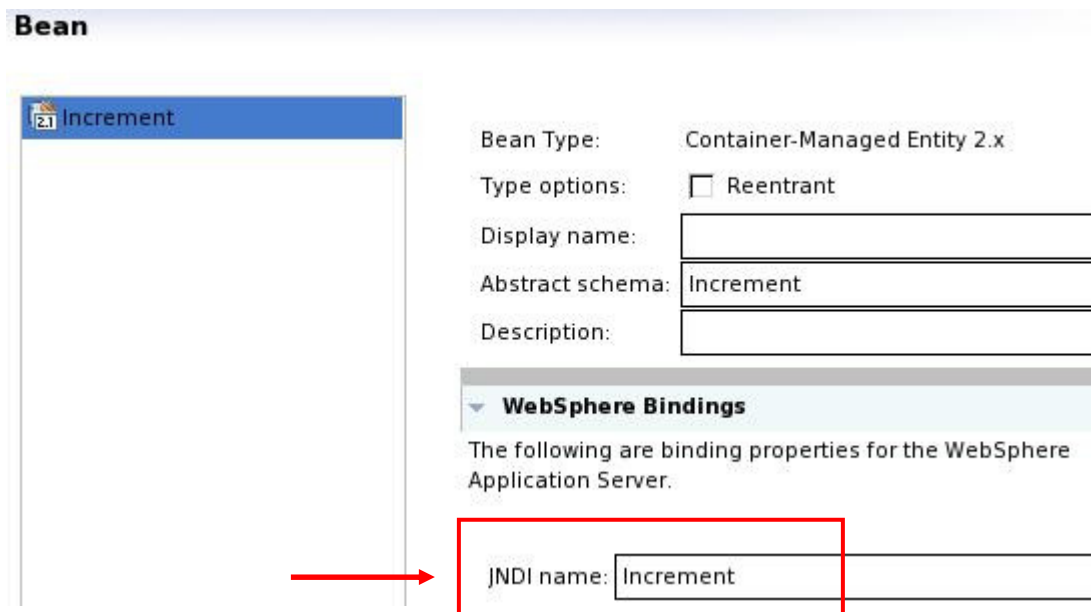


__ i. Double-click on the **HitCountEJB** deployment descriptor to open the graphical EJB Deployment Descriptor Editor

__ j. In the Enterprise JavaBeans section of the EJB Deployment Descriptor, click on **Increment**



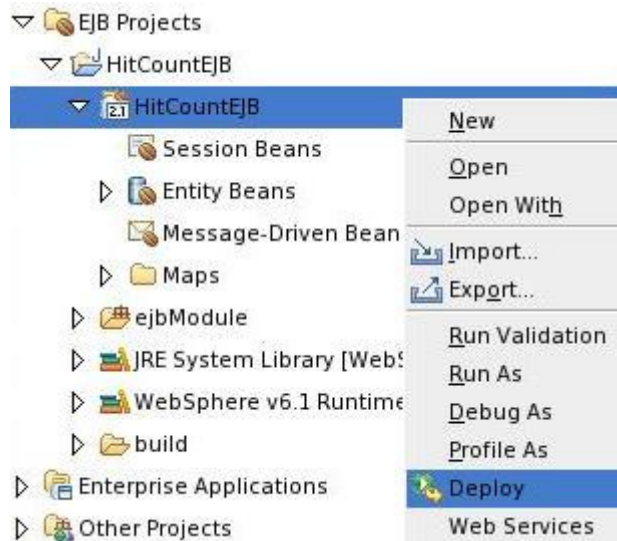
__ k. In the WebSphere Bindings section of the editor, change the JNDI name to **Increment**



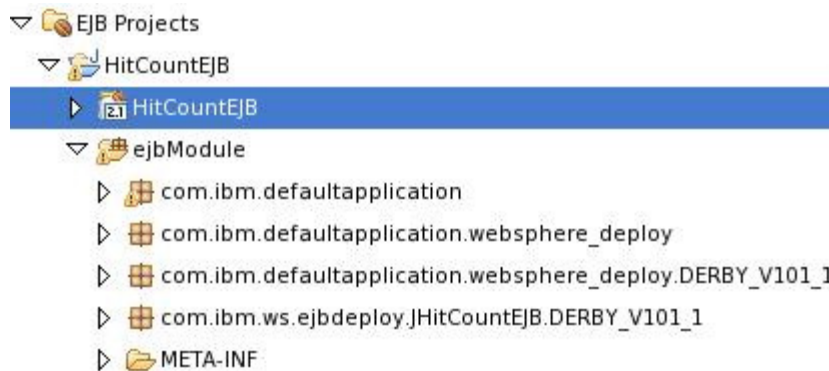
__ l. Use **CTRL-S** to save the EJB Deployment Descriptor

__ m. **Close** the EJB Deployment Descriptor editor

- ___ n. Returning to the Project Explorer view, ensure the **HitCountEJB** deployment descriptor is still selected. Right-click and choose **Deploy**



- ___ o. Expand the **ejbModule** package. This is the location of the source code for the Increment Bean and the generated code from the **Deploy** step. Ignore any warnings from the deployment code generation.



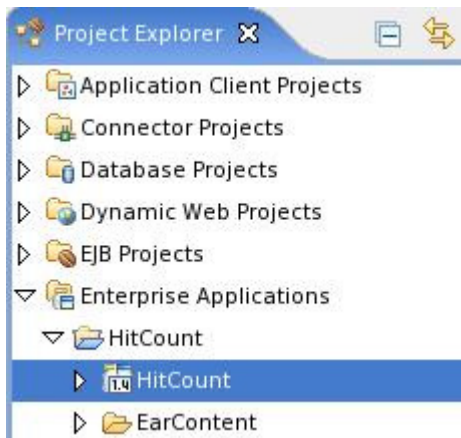
The EJB Project is now complete and ready for deployment and testing. You will perform these tasks in the next section.

Part 5: WebSphere Enhanced EAR support

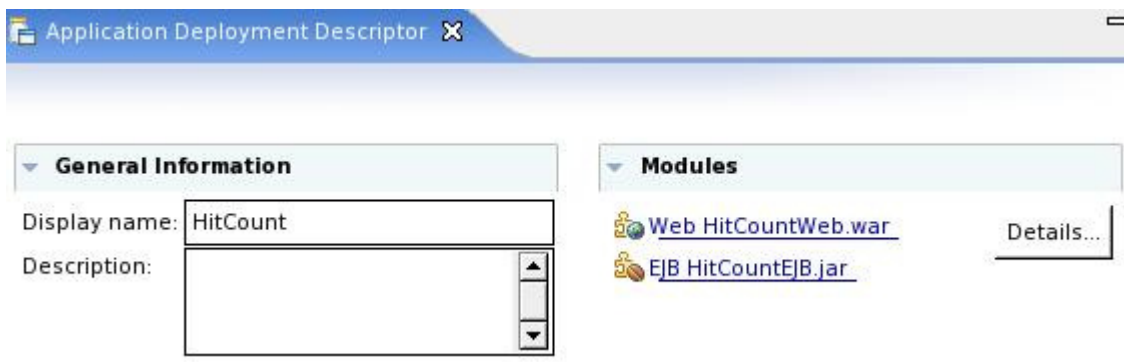
The Application Server Toolkit v6.1 has support for the WebSphere Enhanced EAR capability, which is used for packaging and preparing applications for publishing to a WebSphere Application Server v6.x.

The WebSphere Enhanced EAR is the deployment page of the Enterprise Application Deployment Descriptor editor. It has been updated to allow you to add resource adapters and connection factories to an enterprise application targeted for WebSphere Application Server v6.x.

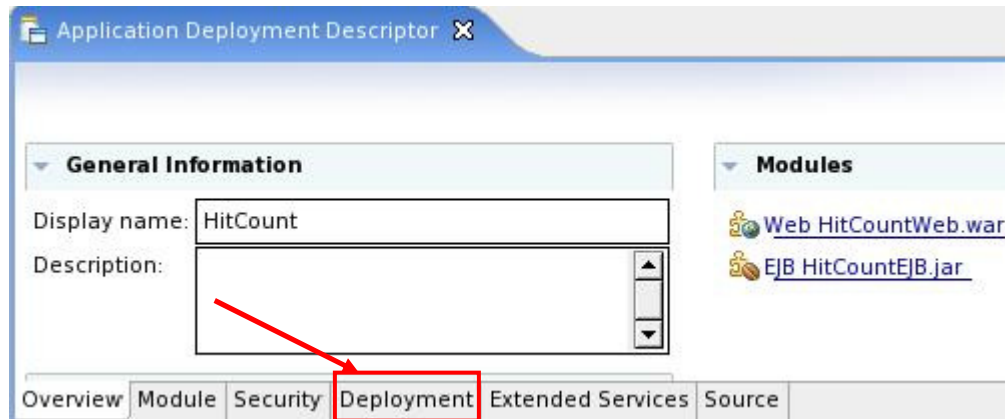
- ___ 1. Configure the HitCount application to use a Data Source and Cloudscape database
 - ___ a. From the Project Explorer view, expand **Enterprise Applications** → **HitCount** and select the **HitCount** application deployment descriptor.



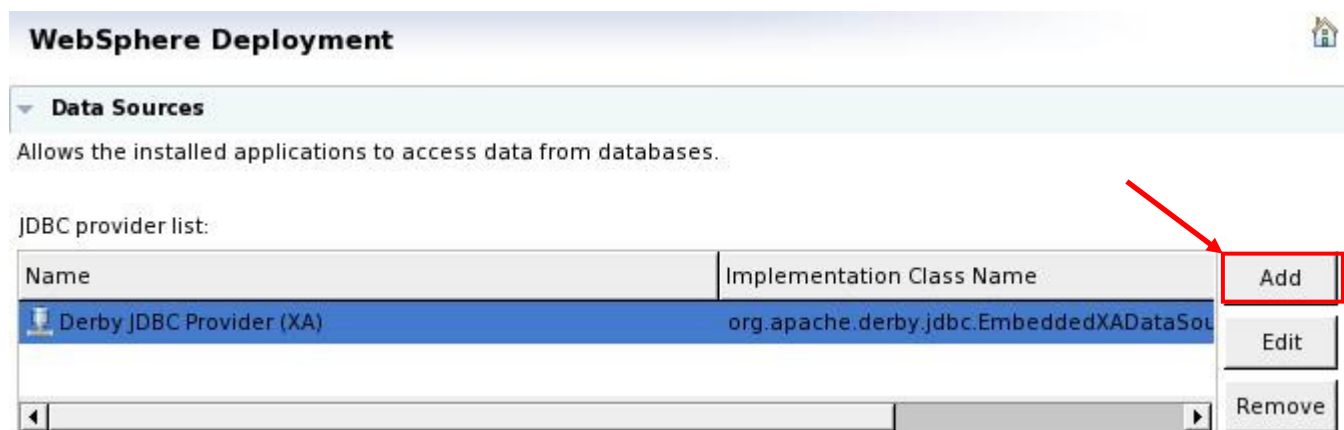
- ___ b. Double-click on the **HitCount** application deployment descriptor to start the Application Deployment Descriptor graphical editor.



___ c. In the Application Deployment Editor, at the bottom of the view, click on the **Deployment** tab

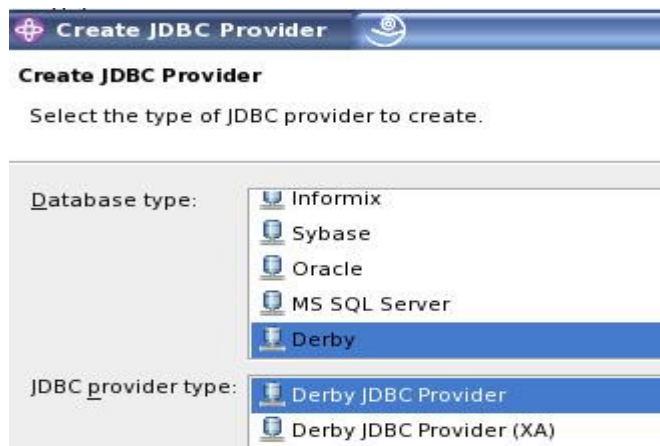


___ d. In the **JDBC provider list** section, click on the **Add** button




___ e. In the **Create JDBC Provider** panel:

- 1) Select a Database type of **Derby**
- 2) Select a JDBC Provider type of **Derby JDBC Provider**
- 3) Click **Next**



___ f. Enter a Name of **Derby JDBC Provider** (use copy and paste from the Description), click **Finish**



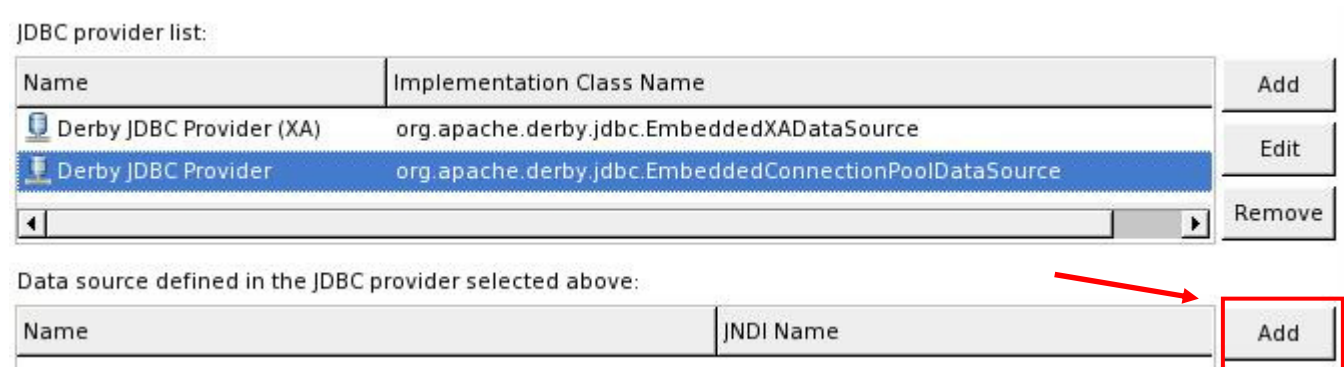
Create JDBC Provider

Select the type of JDBC provider to create.

Name: Derby JDBC Provider

Description: Derby JDBC Provider

___ g. With the **Derby JDBC Provider** selected in the JDBC Provider list, click the **Add** button in the **Data source defined** section.



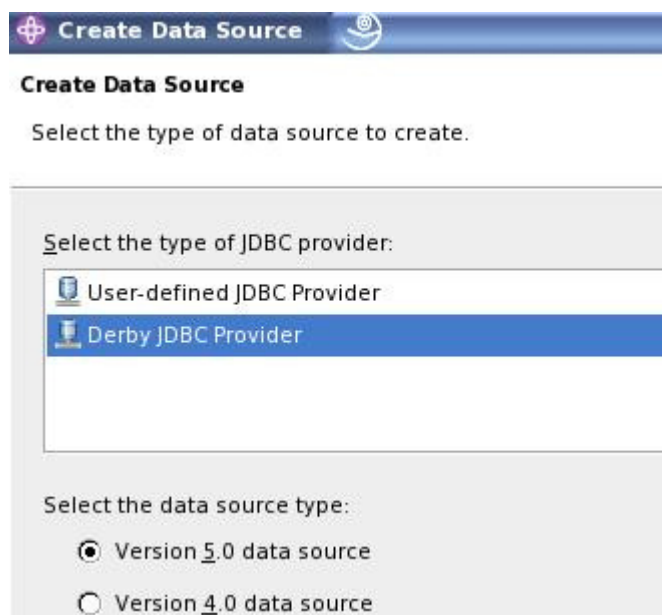
JDBC provider list:

Name	Implementation Class Name	
Derby JDBC Provider (XA)	org.apache.derby.jdbc.EmbeddedXADataSource	Add
Derby JDBC Provider	org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource	Edit
		Remove

Data source defined in the JDBC provider selected above:

Name	JNDI Name	Add
------	-----------	-----

___ h. Select a Data Source of **Derby JDBC Provider**, click then click **Next**



Create Data Source

Select the type of data source to create.

Select the type of JDBC provider:

- User-defined JDBC Provider
- Derby JDBC Provider**

Select the data source type:


- ☒ Version 5.0 data source
- ☐ Version 4.0 data source

___ i. In the Data Source properties section, change the **JNDI Name** to `jdbc/Default` and click **Next**

Create Data Source






Select the type of data source to create.

Name: Data source 1

JNDI name:  jdbc/Default

___ j. In the Resource Properties section, with the **databaseName** selected, enter a value of `/root/astlabDB`. Click **Finish**


Resource Properties:

Name	Description
 databaseName	adminRequired=true - This is a required property.
 shutdownDatabase	If set to the string 'shutdown', this will cause the d
 dataSourceName	Name for ConnectionPooledDataSource. Not used
 description	Description of the Data Source. Not used by the D
 connectionAttributes	Connection attributes specific to Cloudscape. Plac

Name: databaseName

Type: java.lang.String



Required: Yes

 Value: /root/astlabDB


Description: adminRequired=true - This is a required property. This property m

___ k. The JDBC provider list and Data source should now appear as shown below:

JDBC provider list:

Name	Implementation Class Name
 Derby JDBC Provider (XA)	org.apache.derby.jdbc.EmbeddedXADataSource
 Derby JDBC Provider	org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource

Data source defined in the JDBC provider selected above:

Name	JNDI Name
 Data source 1	jdbc/Default

Resource properties defined in the data source selected above:

Name	Value
 databaseName	/root/astlabDB

___ l. Use **CTRL-S** to save your changes.

___ m. Close the **Application Deployment Descriptor** editor.

Part 6: Configuring the WebSphere test environment

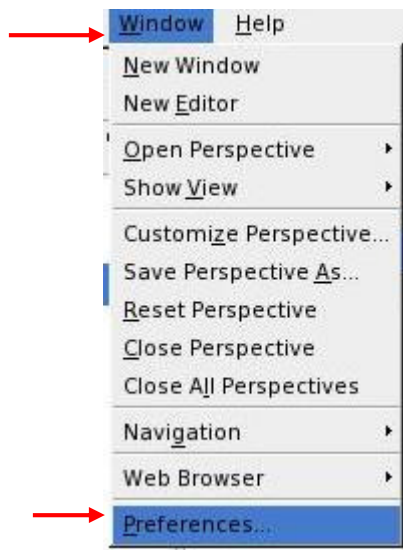
The *WebSphere® test environment* is a runtime environment that is integrated into the workbench for testing applications that are targeted for WebSphere Application Server.

The test environment for WebSphere Application Server v6.1 requires a **full** installation of the WebSphere Application Server and is enabled through a **Run server with resources within the workspace** publishing setting.

For this lab, WebSphere Application Server v6.1 is already installed on your machine and there is no need to install a test environment.

1. Configure and set the WebSphere test environment

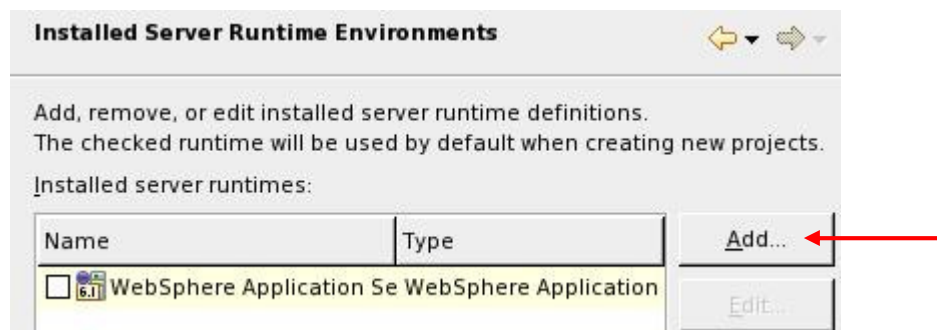
- a. From the Application Server Toolkit menubar, select **Window → Preferences...**



- b. On the left-hand menu, expand **Server**, select **Installed Runtimes**



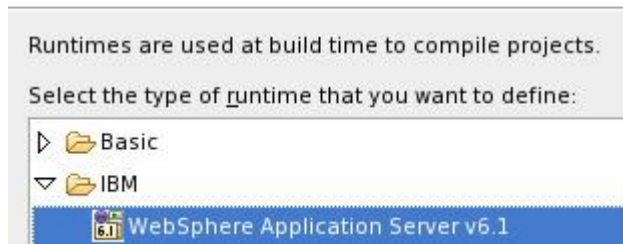
- c. In the **Installed Server Runtime Environments** panel, click **Add...**



- ___ d. In the **New Server Runtime** dialog, accept the default server runtime of WebSphere Application Server v6.1 and click **Next**.

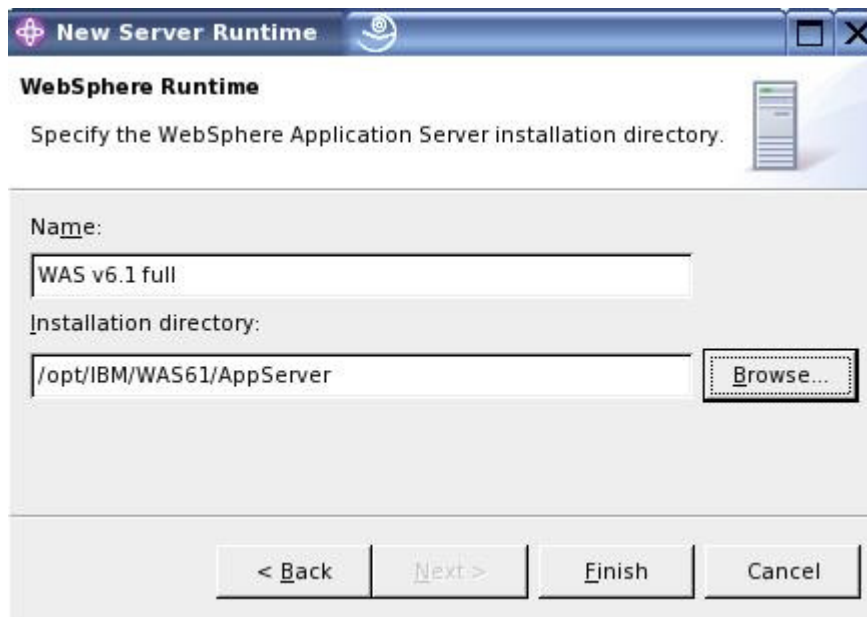
New Server Runtime

Define a new installed server runtime environment

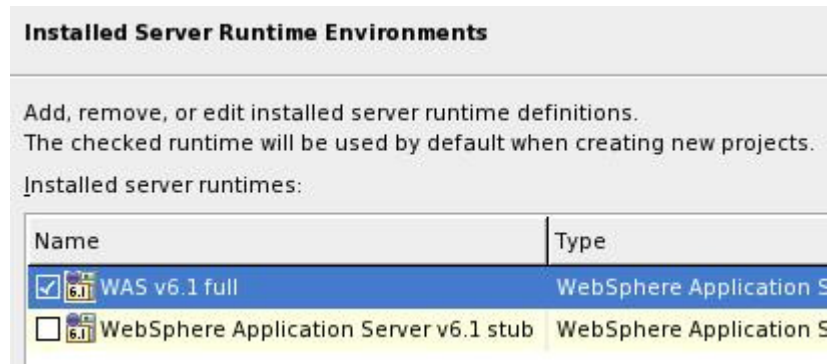


- ___ e. In the WebSphere Runtime panel -

- 1) Enter a **Name** of **was v6.1 full**
- 2) Set the installation directory to **/opt/IBM/WAS61/AppServer** – use the **Browse...** button
- 3) Click **Finish**



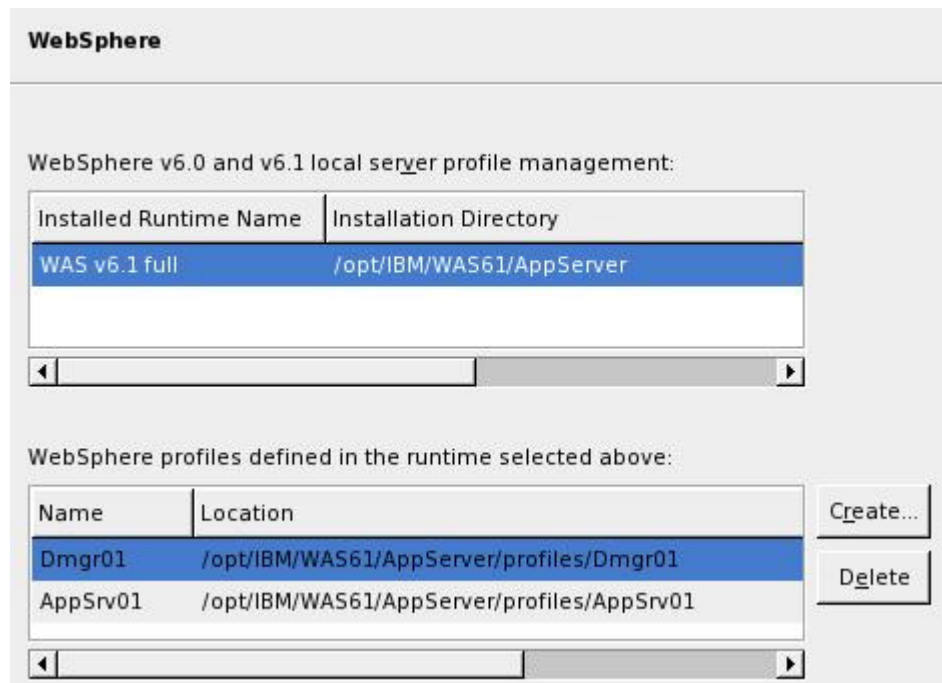
___ f. In the Installed Server Runtime Environments panel, select **WAS v6.1 full** as the default runtime



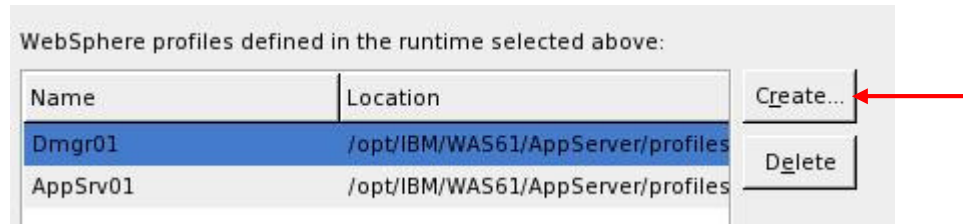
___ g. Return to the left-hand side preferences and select **Server → WebSphere**



___ h. On the right-hand side of the view, the configured WebSphere runtimes and profiles are shown. The **Dmgr01** and **AppSrv01** profiles are part of a cell which you created in Lab01.



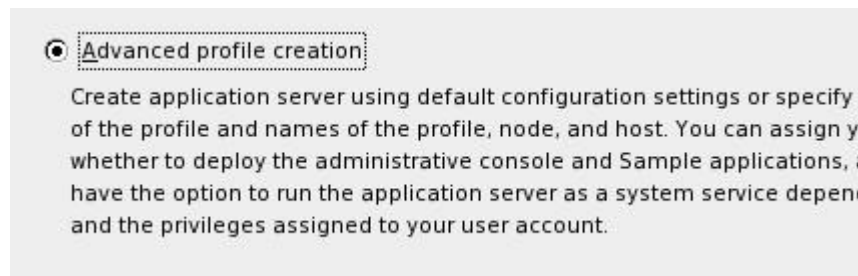
- ___ i. This lab will create a new standalone Application server profile. You can launch the WebSphere Profile Management tool from within the workbench to create profiles. Also, from within the workbench, you can remove the registry and configuration files associated with a profile. Click the **Create...** button to start the Profile Management Tool



- ___ j. After the Profile Management Tool starts, click **Next**
- ___ k. In the Environment Selection panel, accept the default environment of **Application server** and click **Next**.



- ___ l. In the Profile Creation Options panel, select **Advanced profile creation** and click **Next**.



- ___ m. In the Optional Application deployment panel, uncheck **Deploy the default application**. Since this lab is creating the HitCount application, the Default Application does not need to be deployed. Click **Next**

Select the applications to deploy to the WebSphere Application Server environment being created.

☒ **Deploy the administrative console (recommended).**
Install a Web-based administrative console that manages the application server. Deploying the administrative console is recommended, but if you deselect this option, the information center contains detailed steps for how to create the console if it does not exist.

☐ **Deploy the default application.**
Install the default application that contains the Snoop, Hello, and HitCount servlets.

☐ **Deploy the Sample applications.**
Install the Sample applications to use the application server and evaluate the latest technologies. Sample applications are not recommended for deployment to production application server environments.

- ___ n. In the Profile panel, accept the Profile name of AppSrv02 and profile directory.

Specify a profile name and directory path to contain the files for the run-time environment. Click **Browse** to select a different directory.

Profile name:
AppSrv02

Profile directory:
/opt/IBM/WAS61/AppServer/profiles/AppSrv02

- ___ o. Accept the Node name and Host name. Click **Next**.

- ___ p. **Disable** Administrative Security. Click **Next**

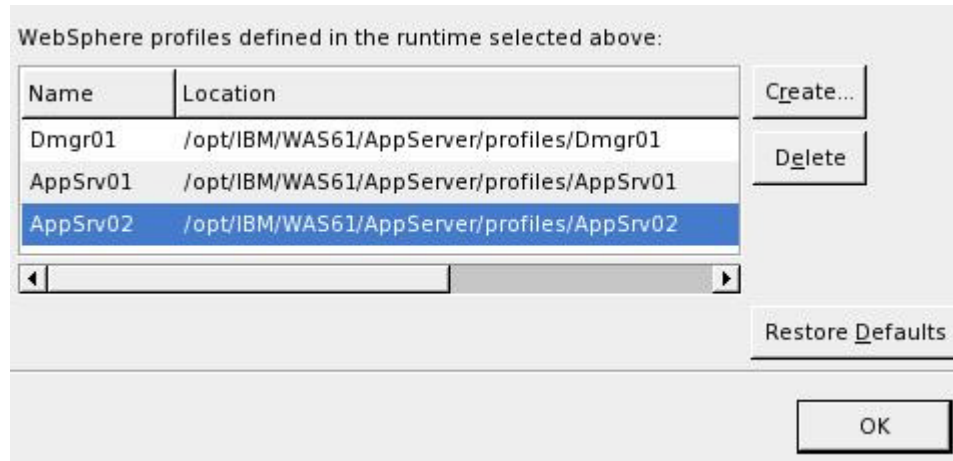
Choose whether to enable administrative security. When you enable administrative security, you can add more users, groups, or external authentication modules. This administrative user is created when you finish. If you do not enable administrative security, you cannot add more users, groups, or external authentication modules.

☐ **Enable administrative security**

- ___ q. Accept the Port Values. Click **Next**

- ___ r. Do not run the application server as a service. Click **Next**

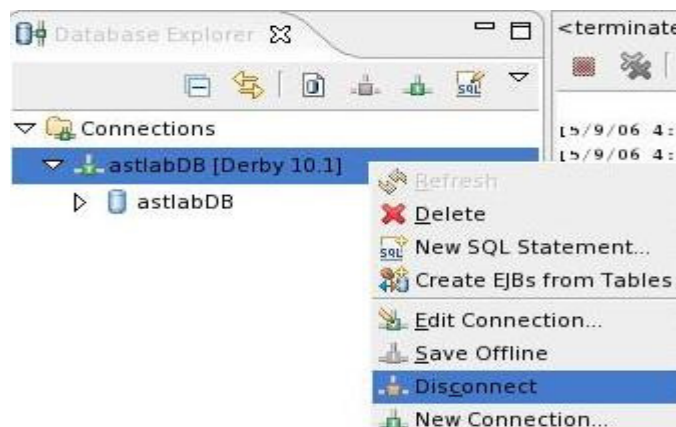
- ___ s. Do not create a Web server definition. Click **Next**
- ___ t. Click **Create**
- ___ u. When the profile creation finishes, **uncheck** 'Launch the First steps console'. Click **Finish**
- ___ v. The **AppSrv02** profile is now listed as a member of the WebSphere profiles. Click **OK**



- ___ 2. In preparation for using the Cloudscape 10.1 (Derby) database with the WAS v6.1 runtime, disconnect from Cloudscape.
 - ___ a. From the perspective bar, switch to the **Data** perspective

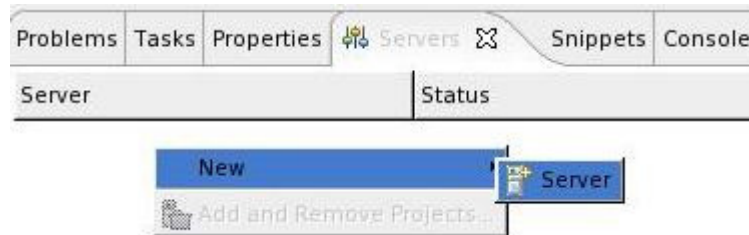


- ___ b. From the **Database Explorer** view, select the **astlabDB [Derby 10.1]** connection. Right-click and select **Disconnect**.

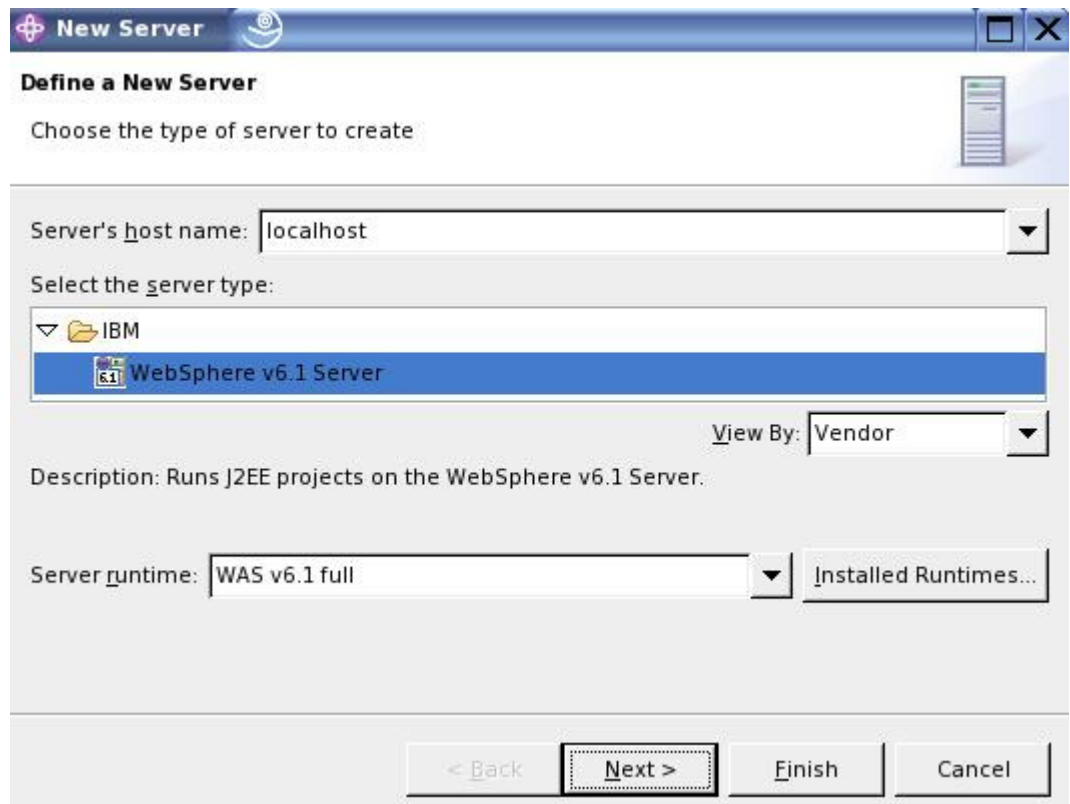


3. **Add a new WebSphere v6.1 test server**

- ___ a. From the perspective bar, switch back to the **J2EE** perspective
- ___ b. Locate and select the **Servers** view. The Servers view allows you to manage the servers. This view displays a list of all your servers and configurations that are associated with that server. You can use this view to start, start in debug mode, start in profile mode, restart, or stop the servers.
- ___ c. Right-click in the Servers view and select **New → Server**



- ___ d. In the **Define a New Server** panel, ensure the server type is **WebSphere v6.1 Server** and the Server Runtime is **WAS v6.1 full**. Click **Next**.



___ e. In the WebSphere Server Settings:

- 1) Select a WebSphere profile name of **AppSrv02**. Notice the tool automatically determines the correct WebSphere Administration port of 2811.
- 2) **Uncheck** 'Security is enabled on this server'. Also notice that the tool is designed to work with Base and Express servers, as well as a Network Deployment topology.
- 3) Click **Next**

New Server

WebSphere Server Settings
Input settings for the new WebSphere server.

WebSphere profile name: AppSrv02

Server connection type and admin port:

☒ **RMI** (Designed to improve communication with the server)

ORB bootstrap port: 2811

☐ **SOAP** (Designed to be more firewall compatible)

SOAP connector port: 8882

☒ Run server with resources within the workspace

☐ **Secu**rity is enabled on this server

Current active authentication settings:

User ID:

Password: ****

Server name: server1

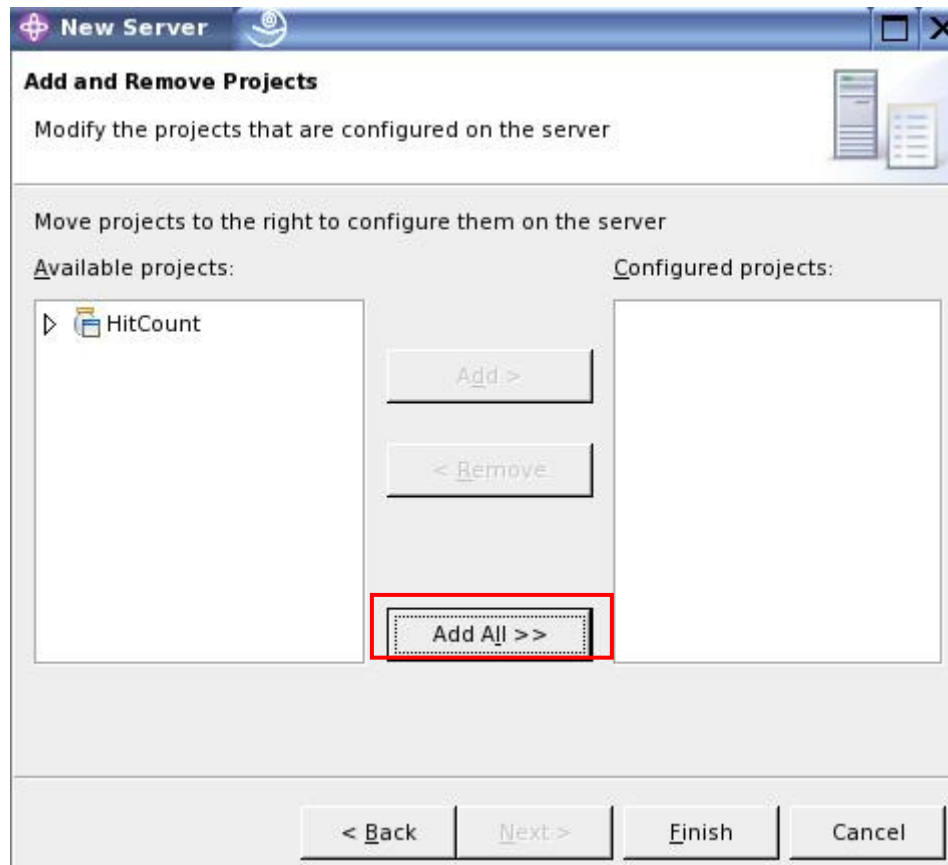
Server type:

☒ **BASE**, Express or unmanaged Network Deployment server

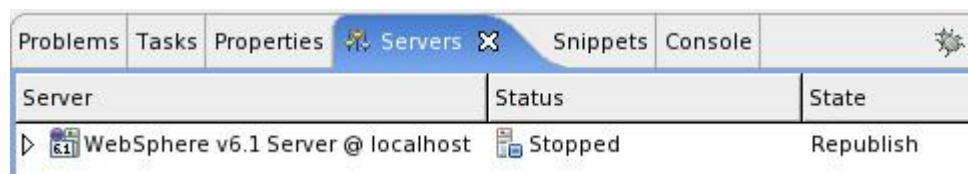
☐ **Net**work Deployment server

___ f. In the **Add and Remove Projects** panel

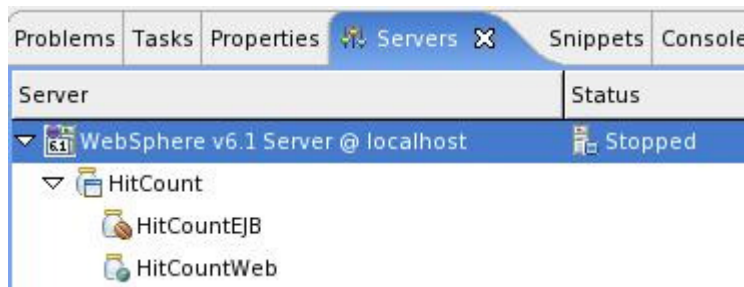
- 1) Click **Add All >>** to add the **HitCount** project to the configured projects for the server.
- 2) Click **Finish** when you are done.



___ g. In the **Servers** view, you should now have a WebSphere v6.1 Server in the Stopped state.



- ___ h. Expanding **WebSphere v6.1 Server @ localhost** shows the HitCount application configured on the server



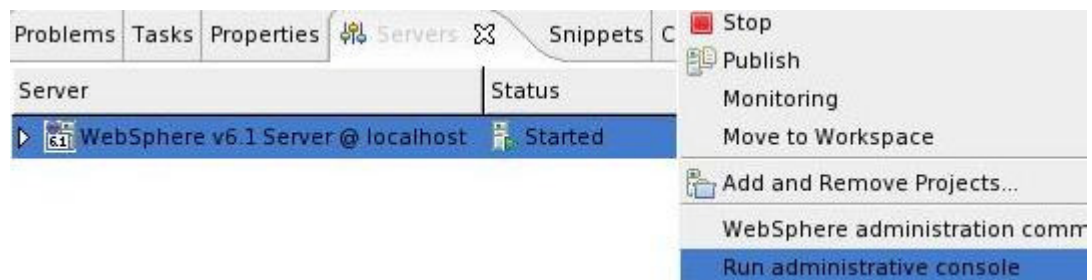
- ___ i. Highlight **WebSphere v6.1 Server @ localhost**. Click the **Start** icon.



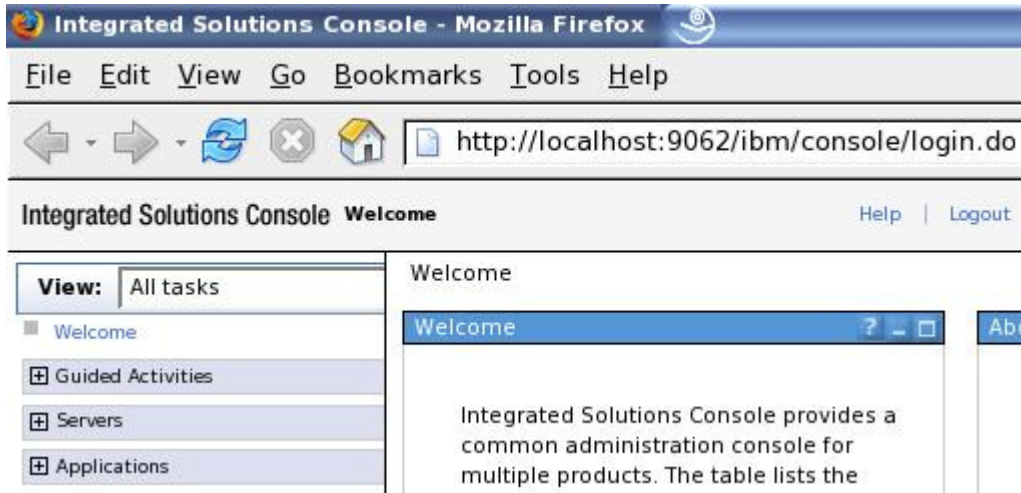
- ___ j. Examine the **Console View** while the server starts.



- ___ k. When the server status has changed to Started and Synchronized, highlight **WebSphere v6.1 Server @ localhost**. Right-click and select **Run administrative console**



___ l. The browser provides access to the WebSphere Application Server Administration Console



___ m. You will use the Administration Console later in the lab. For now, **close** the browser.

Part 7: Testing enterprise beans in the Universal Test Client

The server tools provide you with a Web-based Universal Test Client where you can test your enterprise beans and Java™ classes.

___ 1. Test the Increment EJB with the Universal Test Client

___ a. In the Servers view, highlight **WebSphere v6.1 Server @ localhost**. Right-click and select **Run universal test client**

___ b. The Universal Test Client runs in a web browser. In the Navigation tree, click on **JNDI Explorer**



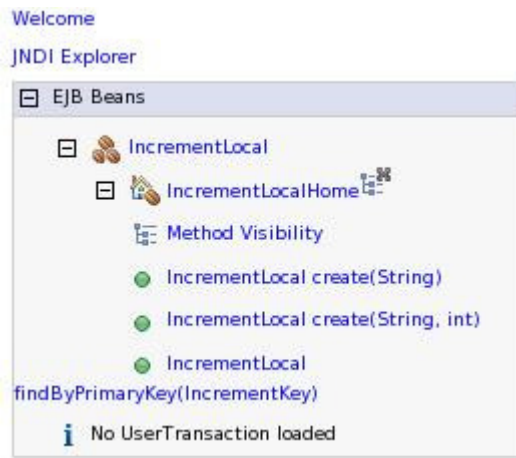
___ c. In the Workspace (the panel on the right), expand **[Local EJB beans]**



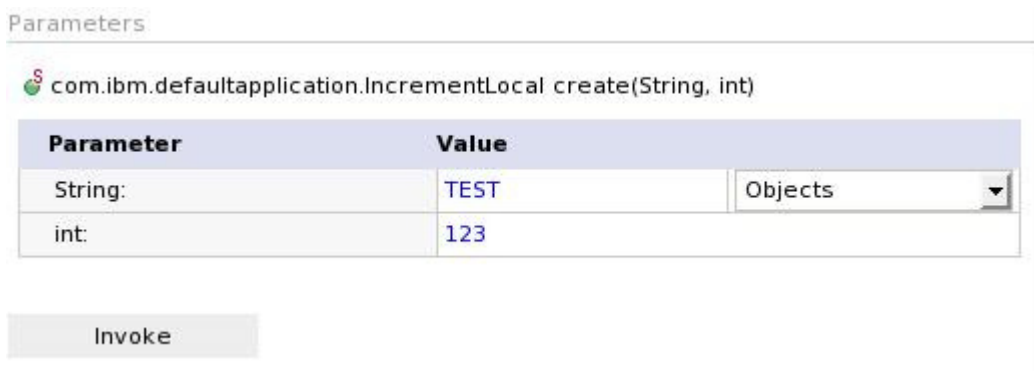
___ d. Click on **Increment (com.ibm.defaultapplication.IncrementLocalHome)**



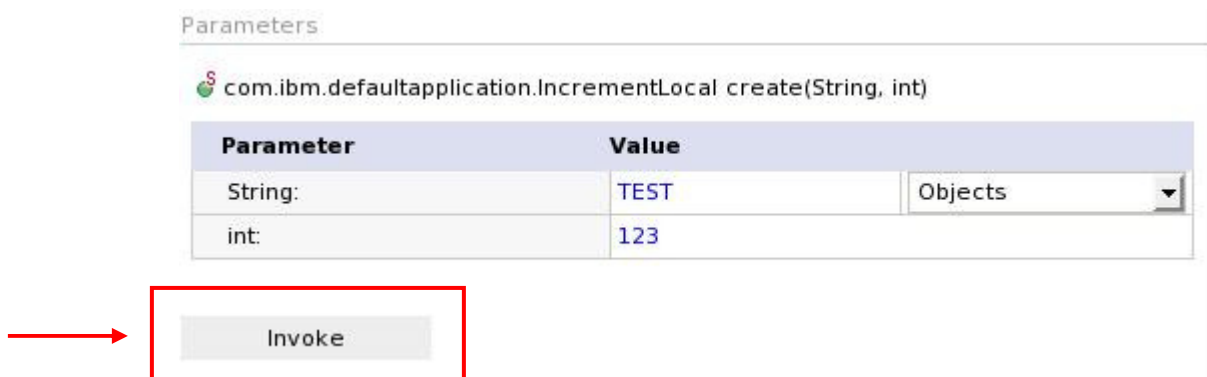
- ___ e. In the Navigation tree, the EJB Beans section will become available. Expand **IncrementLocalHome**



- ___ f. Click on **IncrementLocal create(String, int)**
- ___ g. In the Workspace, enter values of your choice for the **String:** and **int:** parameters. For example, use **TEST** and **123**.



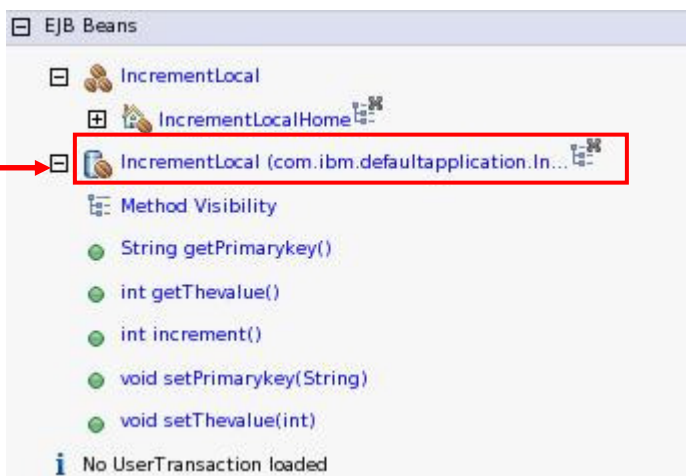
- ___ h. Click the **Invoke** button



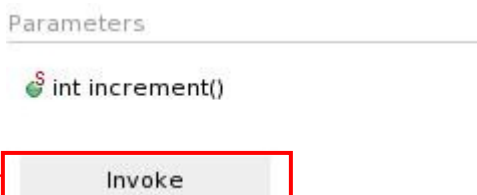
- ___ i. The invocation should succeed. Below the **Invoke** button, there should now be a **Work with Object** button.



- ___ j. Click on the **Work with Object** button.
- ___ k. In the Navigation tree, the EJB Beans section will show the newly created EJB as **IncrementLocal (com.ibm.defaultapplication.Increment)** and the associated methods



- ___ l. Click on the `int increment ()` method
- ___ m. In the Workspace, you can now invoke the `int increment ()` method by clicking the **Invoke** button.




- ___ n. The result should be one greater than the integer value used to create the Increment EJB. In this example, the result is **124** since the object was initially created with **123**.

Parameters

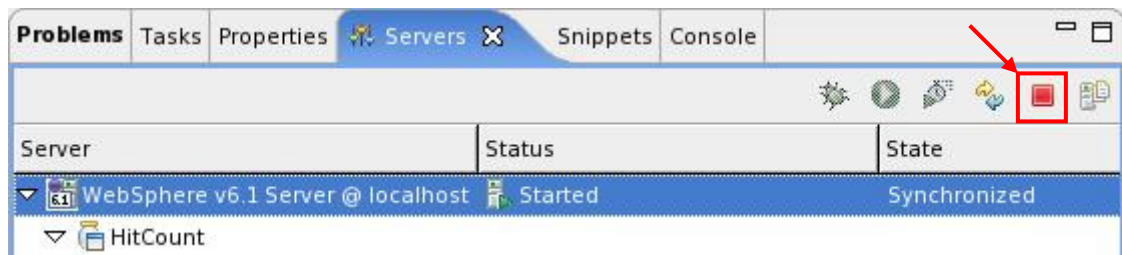
 int increment()

Invoke

▼ Results from  com.ibm.defaultapplication.IncrementLocal.increment()

 124 (int)

- ___ o. Continue to explore the Universal Test Client as desired. When finished, close the browser.
- ___ p. In the Servers view, highlight **WebSphere v6.1 Server @ localhost**. Click the **Stop** icon.



Part 8: Developing Web Applications

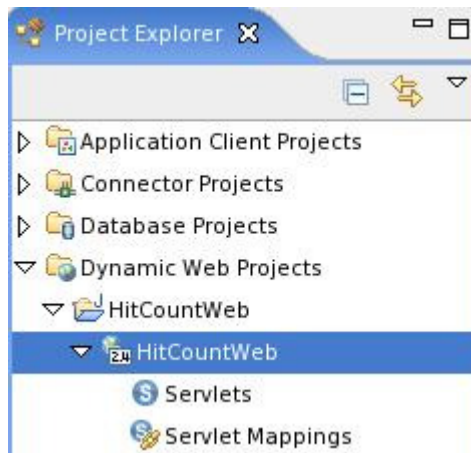
The integrated Web development environment makes it easy to cooperatively create, assemble, publish, deploy and maintain dynamic, interactive Web applications

The WebSphere® Application Server Toolkit allows you to perform tasks for Web applications, such as JSP and HTML file creation, validation, editing, and debugging.

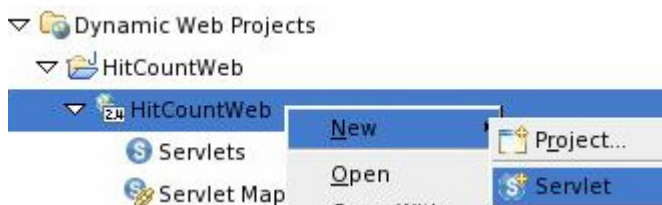
In this section of the lab, you will develop a Servlet and Java Server Page (JSP) to access the Increment EJB.

____ 1. Create and use a Dynamic Web Project

- ____ a. In the AST, from the Project Explorer view, expand the **Dynamic Web Projects** → **HitCountWeb** and select the **HitCountWeb** deployment descriptor.

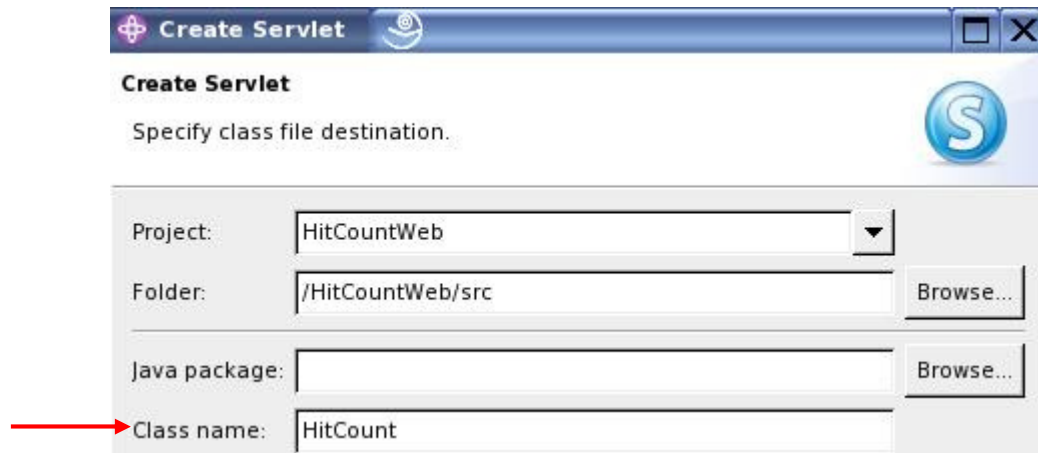


- ____ b. Right-click and select **New** → **Servlet**

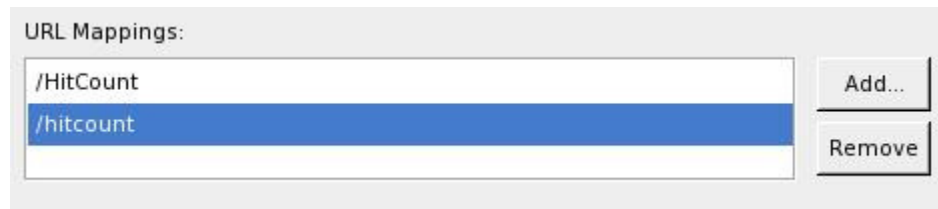


___ c. In the **Create Servlet** panel --

- 1) Enter a Class name of **HitCount**
- 2) Click **Next**

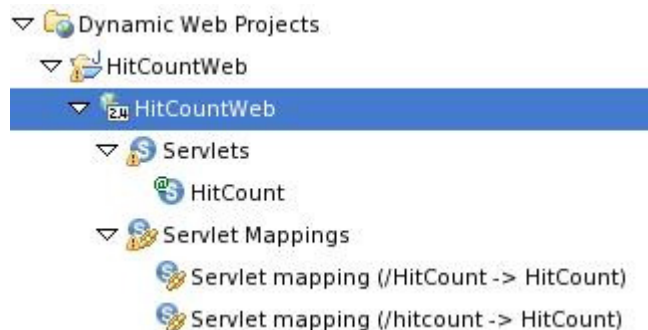


___ d. In the deployment descriptor information, locate the **URL Mappings** section. Click the **Add** button. In the URL mappings popup, enter a URL mapping of `/hitcount` and click **OK**. Click **Next**

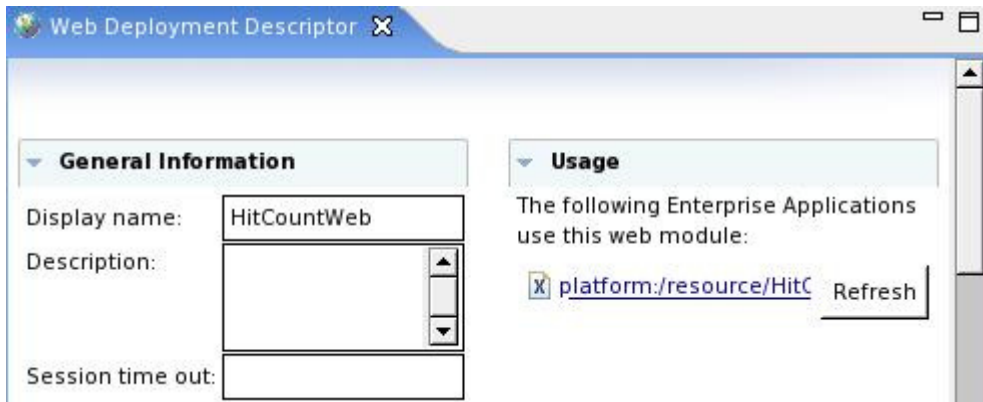


___ e. The final panel allows for the selection of Modifiers, Interfaces, and method stubs. Click **Finish**. Ignore the warning that HitCount does not declare a serialVersionUID field.

___ f. In the Project Explorer, expand the **HitCountWeb** deployment descriptor to see the HitCount Servlet and Servlet Mappings.



- ___ g. The **HitCountWeb** deployment descriptor will still be selected. **Double-click** on it to open the graphical Web Deployment Descriptor editor. You'll use this shortly.



- ___ h. To save time, you will now replace the existing **HitCount servlet** with a version that is completed. You will use copy and paste, similar to the previous example.

- 1) From the SLES Desktop, locate the KDE Panel at the bottom of the workspace. Click on the 'Personal Files' icon.



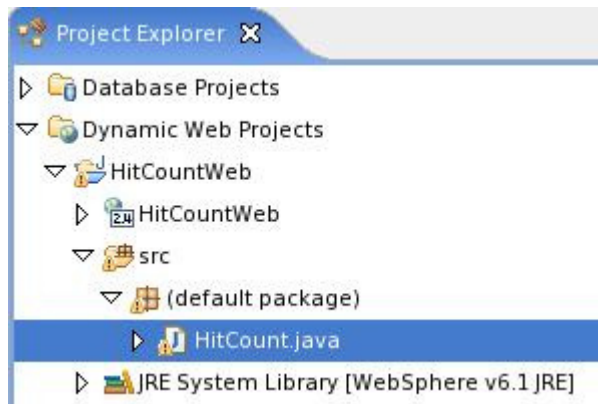
- 2) Navigate to `/root/WAS61STEW/hands-on/Lab02-AST`. You should see three files.



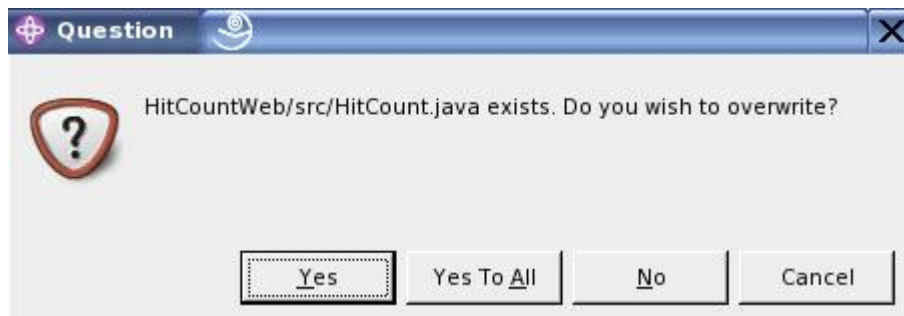
- 3) Right-click on **HitCount.java** and select **Copy**. (you want to copy the entire file, not the contents)



- 4) Return to the AST. Under the HitCountWeb project, expand the **src** folder and the **(default package)** folder. Select **HitCount.java**



- 5) Enter **CTRL+V** (paste). You should get a warning message:



- 6) Select **Yes** to overwrite the file
- 7) The **Web Deployment Descriptor** editor will show a change has been made (the **asterisk** on the view tab)



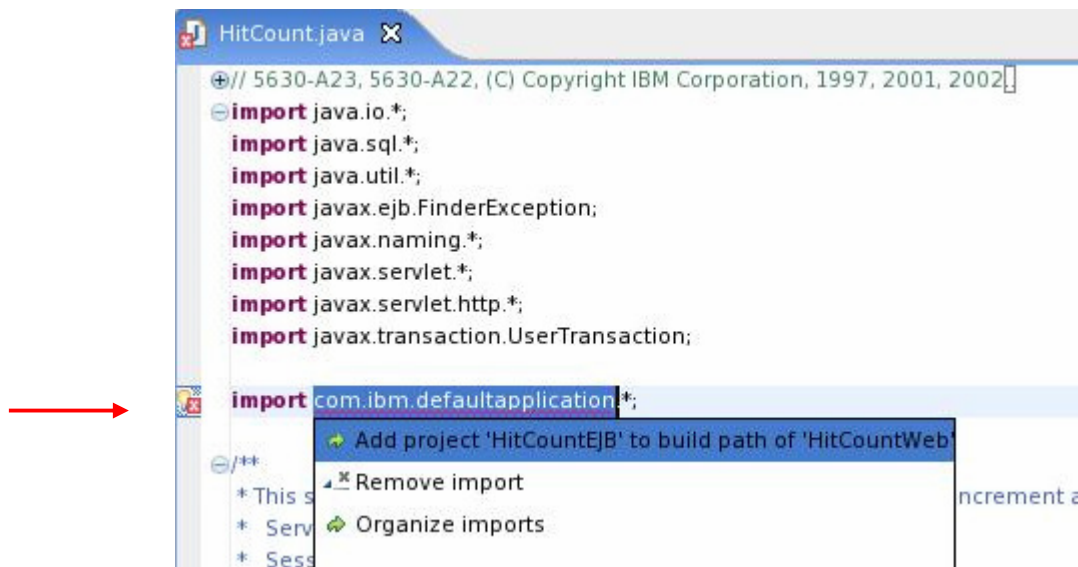
- 8) Close the **Web Deployment Editor**. **Do not save the changes**. Click **No**. This will preserve the existing servlet mappings.



__ i. There are now errors in **HitCount.java**.

- 1) Find the first row with a **red X** beside it. This should be the line:

```
import com.ibm.defaultapplication.*;
```
- 2) Single-click on the **red X** in the left margin to get suggestions on fixing the problem.
- 3) Select **Add project 'HitCountEJB' to build path of 'HitCountWeb'**

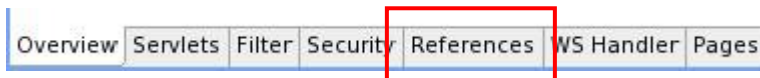


__ j. The errors should now be resolved (a few warnings will remain). Close the **HitCount.java** editor

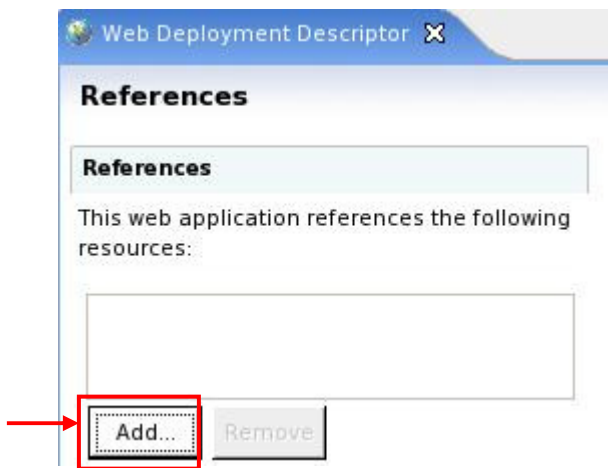
- ___ k. One final requirement of the HitCount servlet is to configure a reference to the Increment EJB. Select the **HitCountWeb** deployment descriptor.



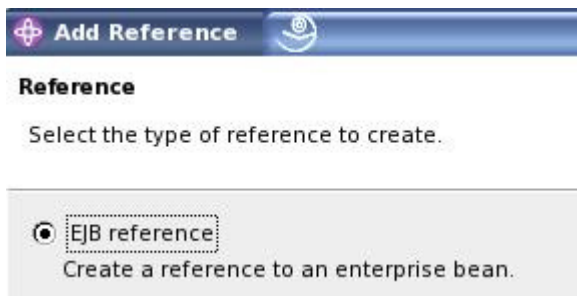
- ___ l. Double-click on **HitCountWeb** to open the graphical Web Deployment Descriptor editor
- ___ m. In the Web Deployment Descriptor editor, at the bottom of the view, select the **References** tab.



- ___ n. In the References section, click **Add...**

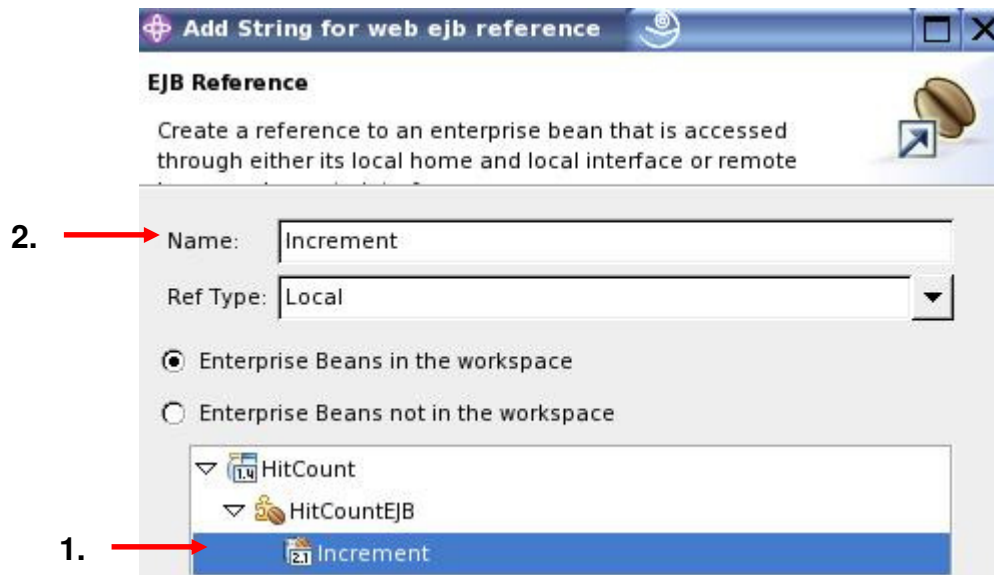


- ___ o. In the Add Reference panel, select **EJB Reference**. Click **Next**



___ p. In the EJB Reference Panel -

- 1) Below the **Enterprise Beans in the workspace** checkbox, expand **HitCount** → **HitCountEJB** and select **Increment**
- 2) Change the **Name** to **Increment**



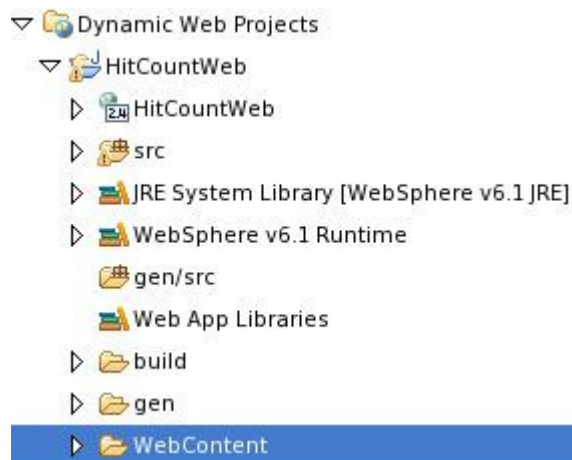
___ q. Click **Finish**

___ r. Use **CTRL-S** to save the file.

___ s. **Close** the Web Deployment Descriptor editor

___ 2. Create a Java Server Page for the HitCount application

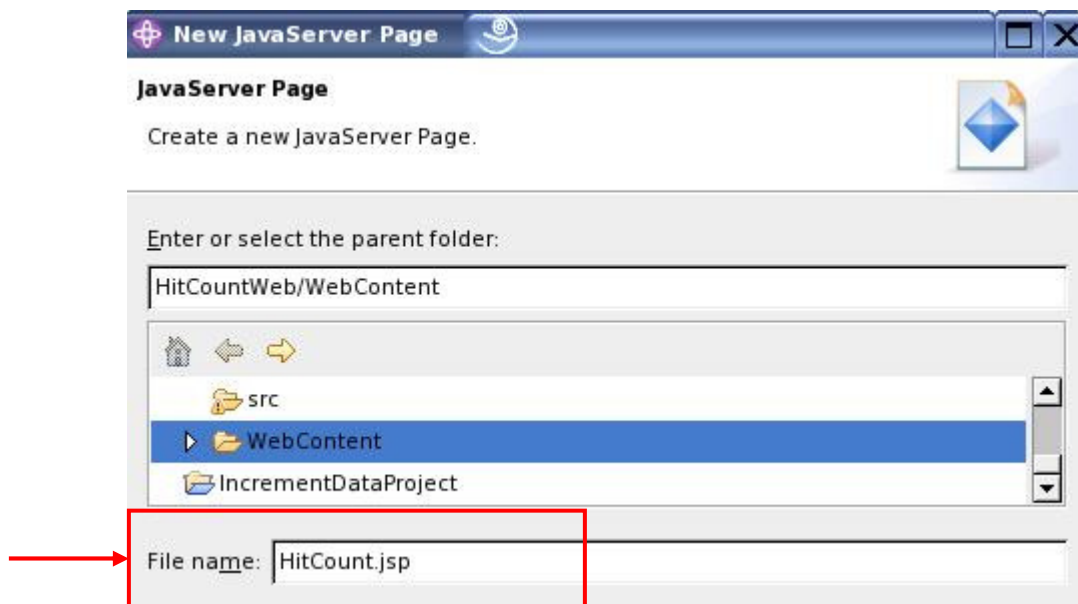
- ___ a. The HitCount application also uses a JSP. Create the framework for the JSP with the creation wizard. From the **HitCountWeb** project, select the **WebContent** folder



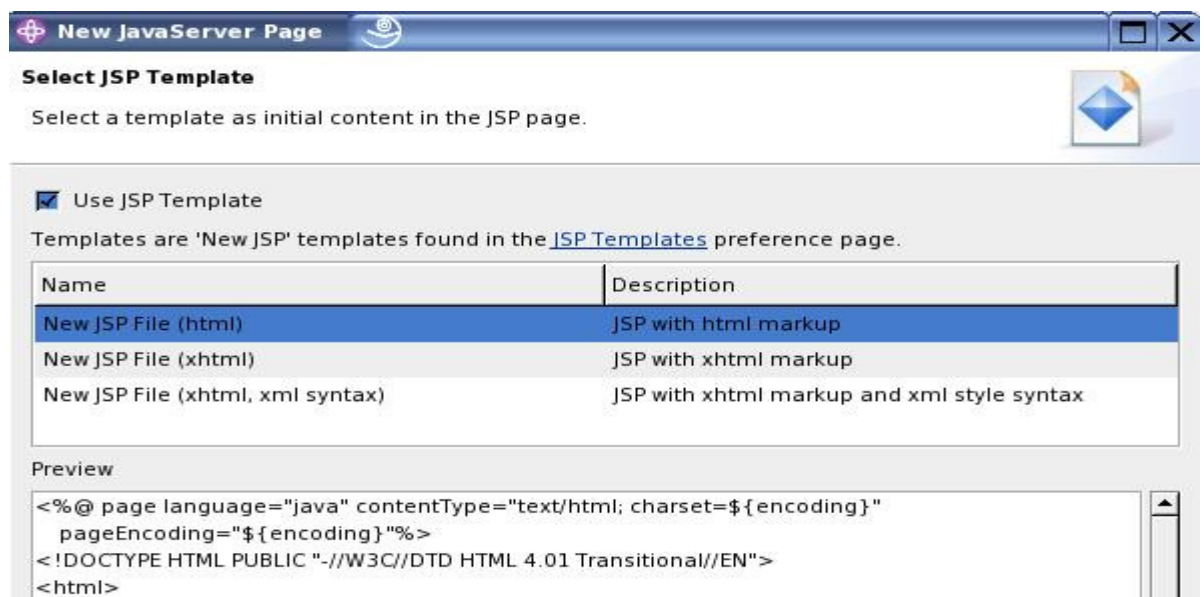
___ b. Right-click on the **WebContent** folder, and select **New → JSP**



___ c. From the New JavaServer Page panel, enter a name of **HitCount.jsp** and click **Next**



___ d. In the JSP template page, take the default of **"New JSP file (html)"** and click **Finish**



- ___ e. The **HitCount.jsp** file is opened in the JSP editor, with a default template. This serves as a starting point for JSP development.



```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
</body>
</html>
```

- ___ f. To save time in the lab, replace the existing **HitCount.jsp** file with a completed version, using the copy and paste techniques used previously.

- 1) From the SLES Desktop, locate the KDE Panel at the bottom of the workspace. Click on the 'Personal Files' icon.



- 2) Navigate to **/root/WAS61STEW/hands-on/Lab2-AST**. You should see three files.



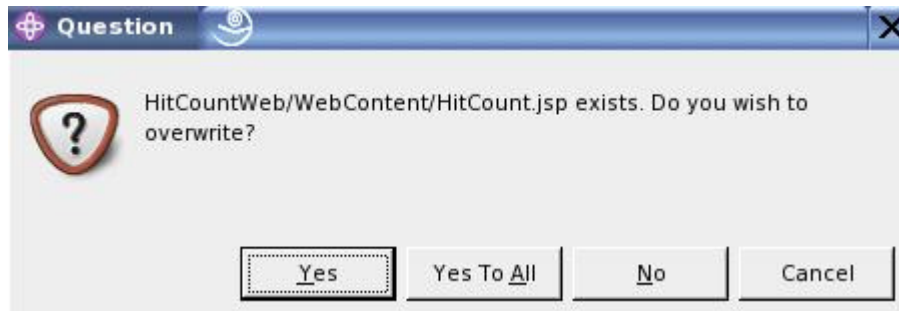
- 3) Right-click on **HitCount.jsp** and select **Copy**. (you want to copy the entire file, not the contents)



- 4) Returning to the AST, expand the WebContent folder and select **HitCount.jsp**



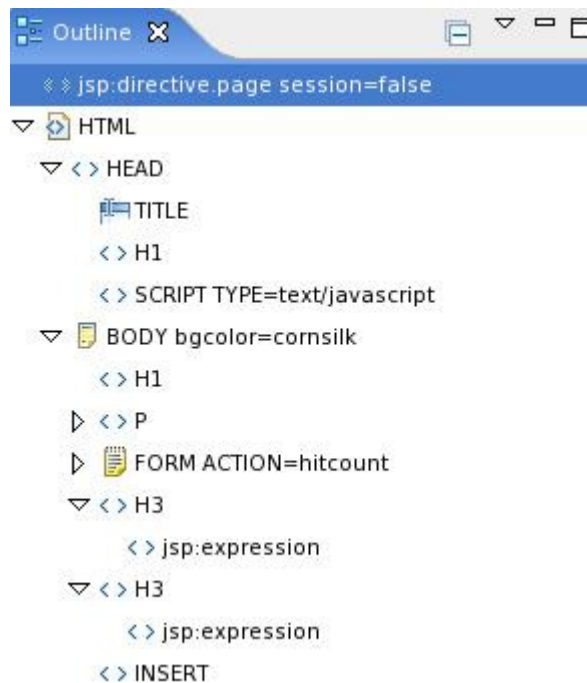
5) Then use **CTRL+V** for paste. You should get a warning message:



6) Select **Yes** to overwrite the file

___ g. The **HitCount.jsp** file is now complete

___ h. In the **Outline** view, you can expand and examine the format of the JSP file



___ i. Close **HitCount.jsp** file editor

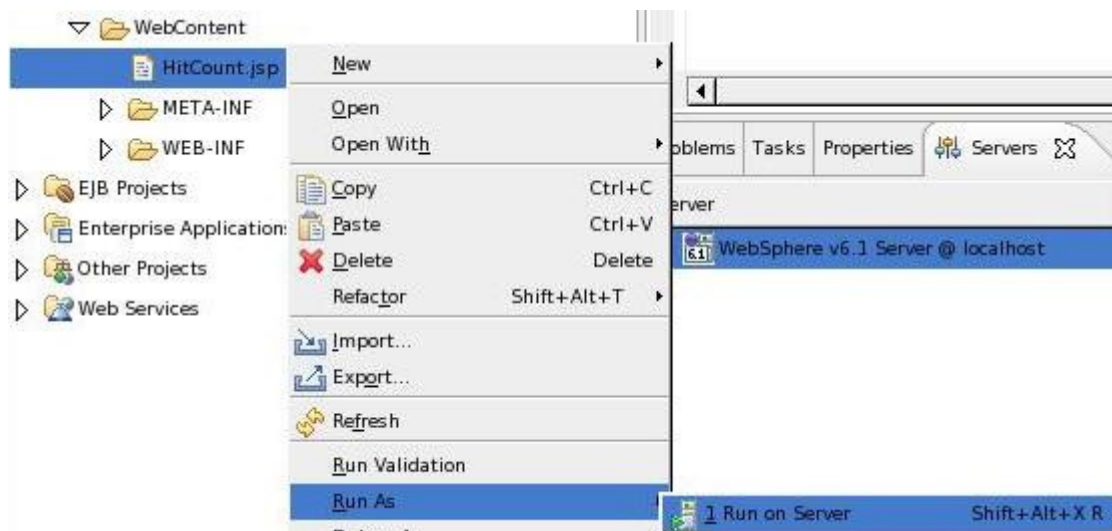
Part 9: Testing the HitCount Application

The testing and publishing tools provides runtime environments where you can test JSP files, servlets, HTML files, enterprise beans and Java™ classes. You can use the workbench to test and publish resources from the following types of projects:

- Dynamic Web projects, which typically contain JSP files, HTML files, servlets, and JavaBeans™
- Static Web projects, which typically contain HTML files and graphic files
- Enterprise Application projects, which may contain Java Archive (JAR) files or Web Archive (WAR) files or both, and pointers to other Web or EJB projects
- EJB projects, which contain enterprise beans

1. Publish and test the HitCount application

a. With **HitCount.jsp** still selected, right-click and select **Run As→ Run On Server**



b. Take the default server selection and select **Finish**.

c. **Wait** for the server to start.

d. The browser starts automatically and accesses the **HitCount.jsp** page with the correct port and Web Context.



- ___ e. Test the HitCount application by selecting any of the values on the HitCount page. For example, select **Enterprise Java Bean (CMP), Local Namespace, and Commit.**

Hit Count Demonstration

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

Select a method of execution:

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

Namespace lookup method for EJB:

- ☐ Global Namespace
- ☒ Local Namespace

Transaction type:

- ☐ None
- ☒ Commit
- ☐ Rollback

Increment

- ___ f. Click the **Increment** button. This should return success and increment the current value. You can further explore the HitCount sample as needed.

Increment

Hit Count value for (Increment EJB): 1

Transaction Commit completed.

Congratulations, you have now finished Lab02.

In this exercise you used the *Application Server Toolkit (AST)* to develop the HitCount application, targeting WebSphere Application Server V6.1. You used the AST wizards and tools for creating new Web applications and EJBs, as well as the tools to edit WebSphere-specific bindings and extensions. The testing and publishing tools provided the runtime environments where you can test your application.