

Using IBM Rational Application Developer Version 6.0 wizards

by Jon Rush
IBM eServer Solutions Enablement

July 2005

Table of contents

Abstract.....	1
Introduction	1
Create a new EJB project	2
Generate EJBs.....	6
Specify a default data source for the EJBs to use.....	24
Create a WebSphere test server environment.....	27
Modify the server configuration for the testing	33
Add a data source or specific back-end iSeries server	51
Test EJBs with test client	59
Summary	68
Appendix A	69
Trademarks.....	70
About the Author	71

Abstract

Using IBM® Rational® Application Developer Version 6.0 makes it easy to create Enterprise JavaBeans™ (EJB™) from back-end IBM DB2 Universal Database™ (UDB) for iSeries™ tables. This technical lab will walk you through using the Rational Application Developer workbench and wizards to generate an entity EJB that will access data in a DB2® UDB for iSeries table for deployment on IBM WebSphere® Application Servers on IBM eServer™ i5 and iSeries systems in the following operating environments: Linux™, IBM Integrated xSeries® Adapters and Integrated xSeries Servers running Microsoft® Windows®, and IBM i5/OS™ and IBM OS/400®. It is intended for DB2 UDB, Rational Application Developer, and WebSphere developers who need to create EJB Web faces for their corporate data repositories.

Introduction

A common business strategy is to enable business decision makers to easily access back-end data through their Web browsers. Giving users access to the data through their Web browsers, which was previously only available through green screens or command lines, enables quicker management decision making and better informed sales staffs and business development personnel.

Done right, Web facing back-end databases can result in immediate return on investment (ROI). Because Web browsers require little or no user training, the costs involved typically equal the resources devoted to Web enabling these databases. Naturally, the easier it is to develop Web-based user interfaces for back-end databases, the lower total cost of ownership (TCO) for such projects, and the faster the ROI.

IBM has several products that minimize TCO and maximize ROI for database Web-facing projects. Rational Application Developer V6.0 is an easy-to-use, powerful application suite for creating EJBs that enable Web browsers to access DB2 UDB for iSeries databases. If these databases reside on WebSphere Application Server for iSeries servers, the result is the perfect storm of usability and performance.

This online lab describes procedures for developing EJB on Rational Application Developer for DB2 UDB for iSeries databases on WebSphere Application Server for iSeries servers.

Create a new EJB project

After you have started Rational Application Developer, create a new EJB project named **EjbRdbiSeriesExample** (as illustrated below).

1. Click **File > New > Project**. This will launch the New Project wizard. (Figure 1)

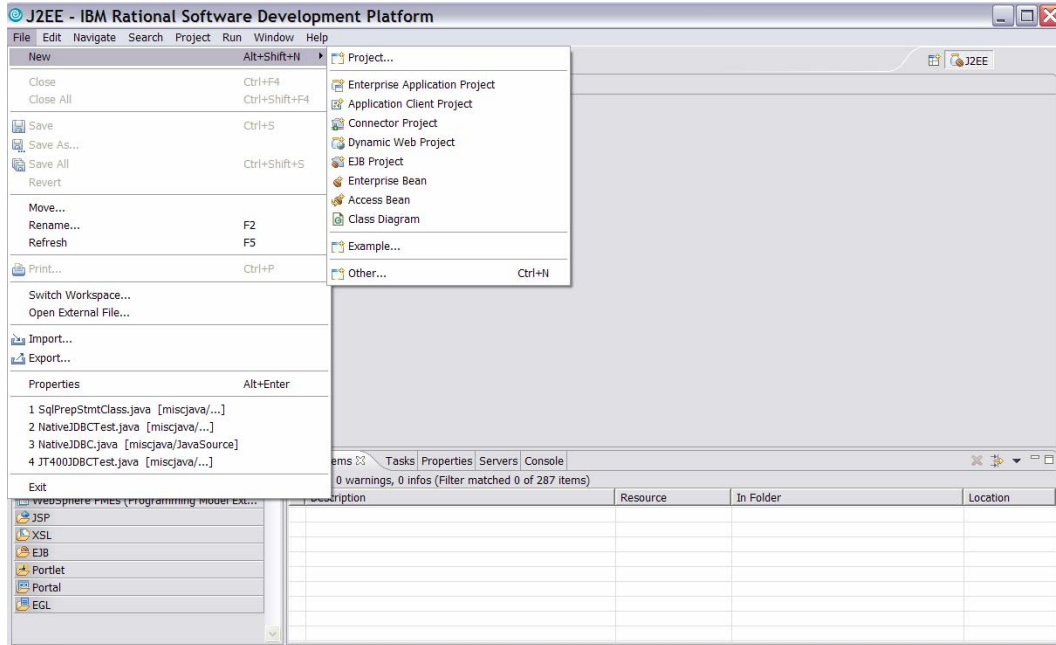


Figure 1: Create a new project

2. Select **EJB Project** from the list of options and click **Next**. (Figure 2)

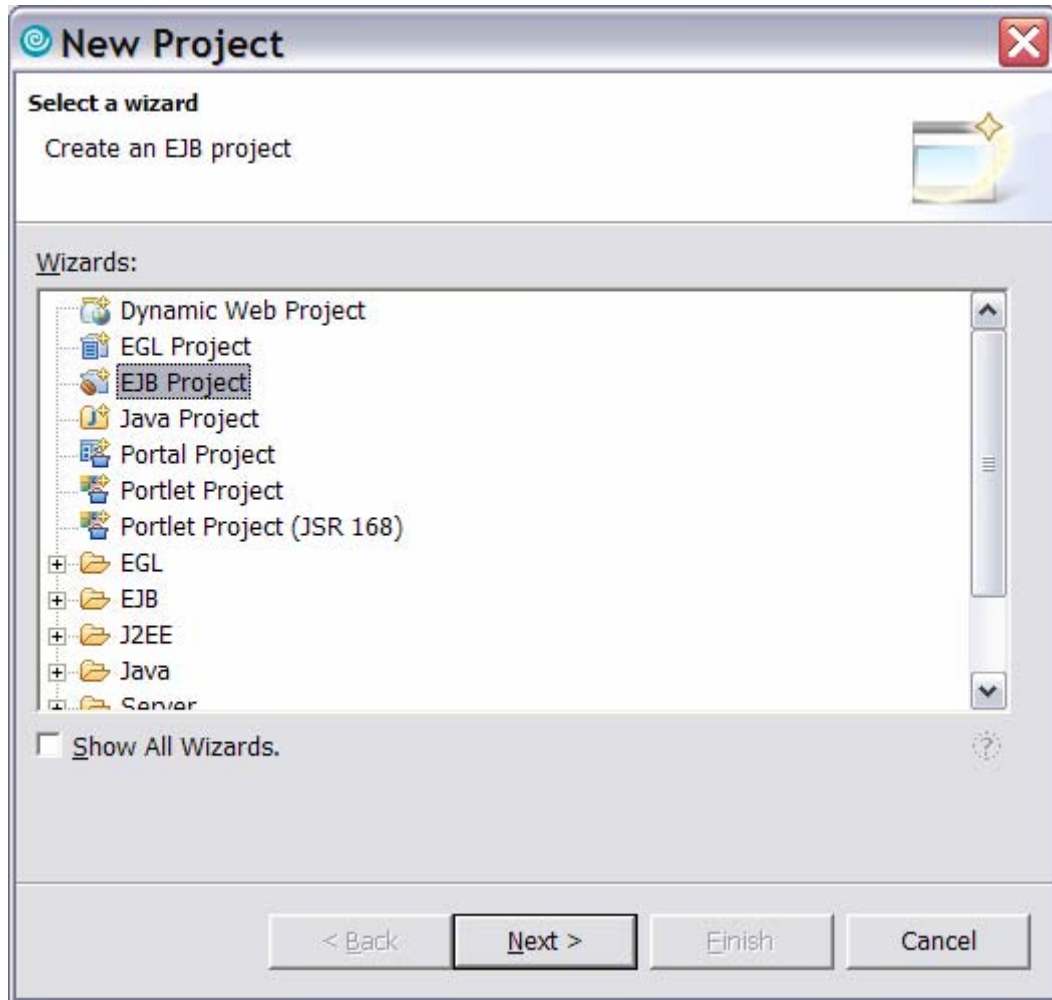


Figure 2: New Project wizard

3. Click **Show Advanced >>**. (Figure 3)

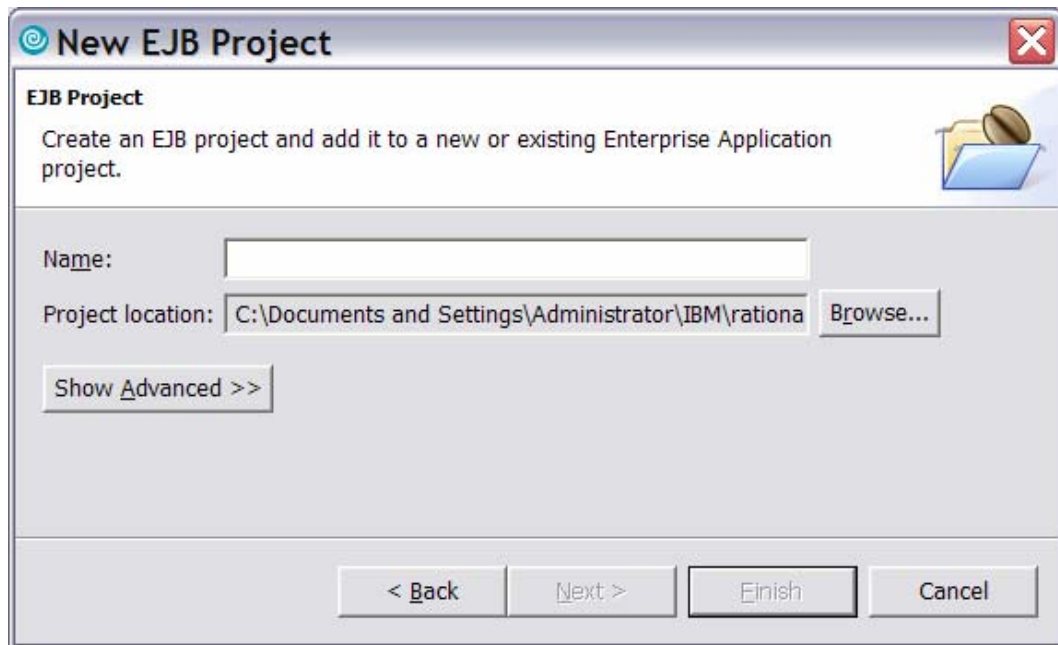


Figure 3: New EJB Project screen

4. Type in the project name: **EjbRdbiSeriesExample**. The EAR project name will automatically be filled in as **EjbRdbiSeriesExampleEAR**. Click **Finish**. (Figure 4)

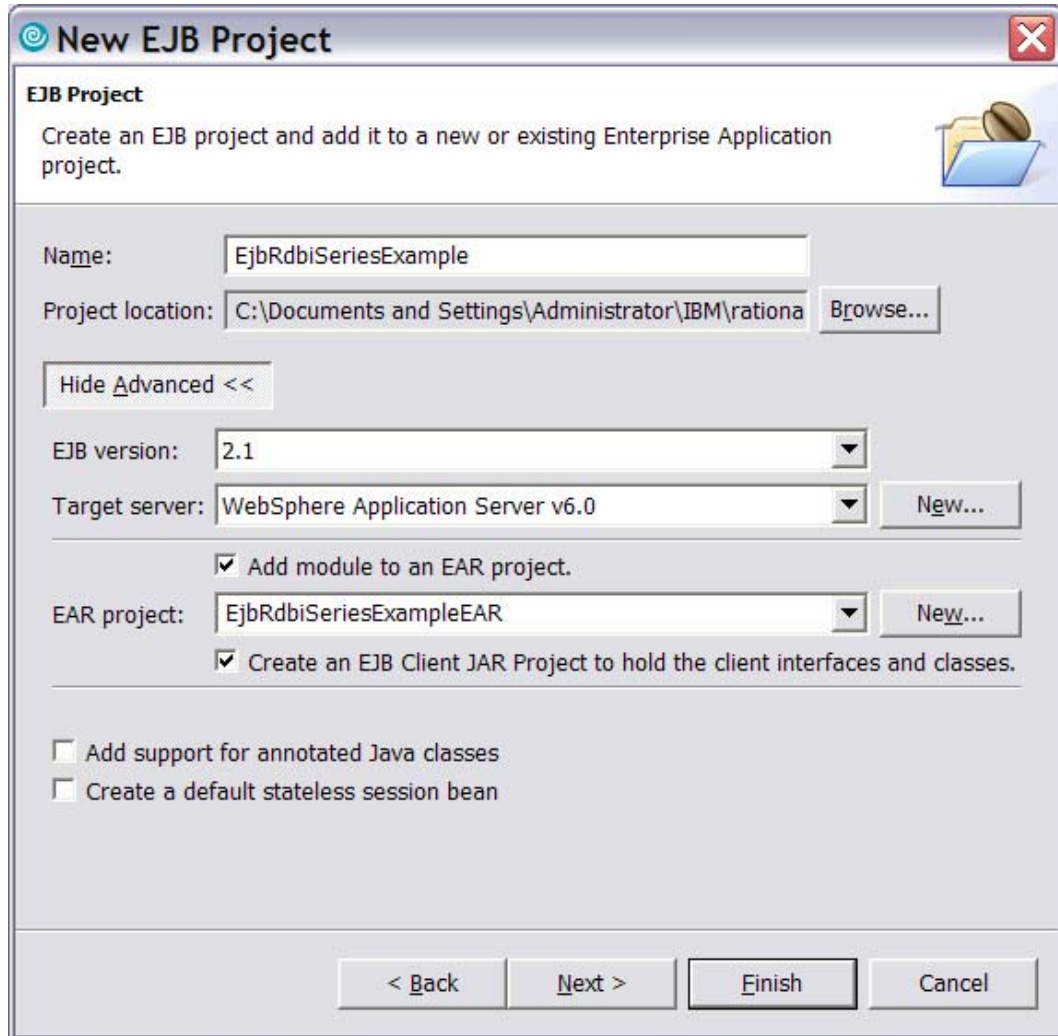


Figure 4: EJB Project finish screen

1. To start the process, right click the **EjbRdbSeriesExample** project and select **EJB to RDB mapping > Generate Map**. (Figure 6)

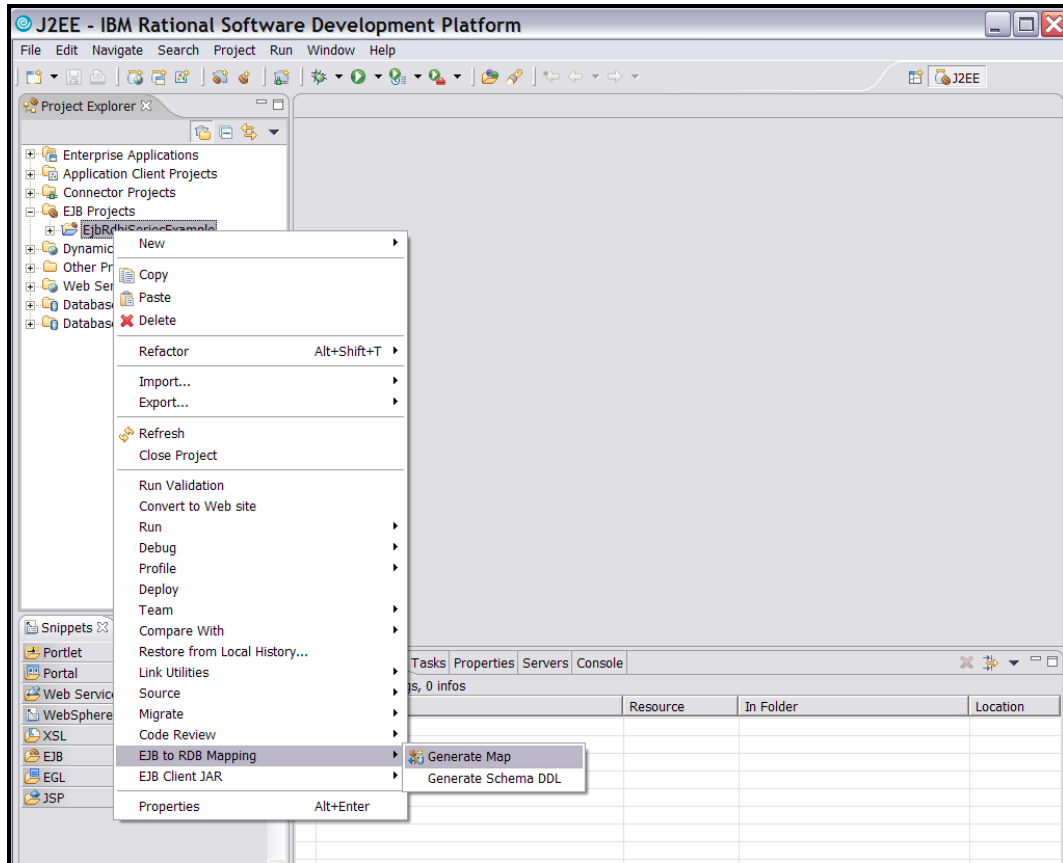


Figure 6: Generate EJB to RDB map

2. Next, select **Create a new backend folder** and click **Next**. (Figure 7)

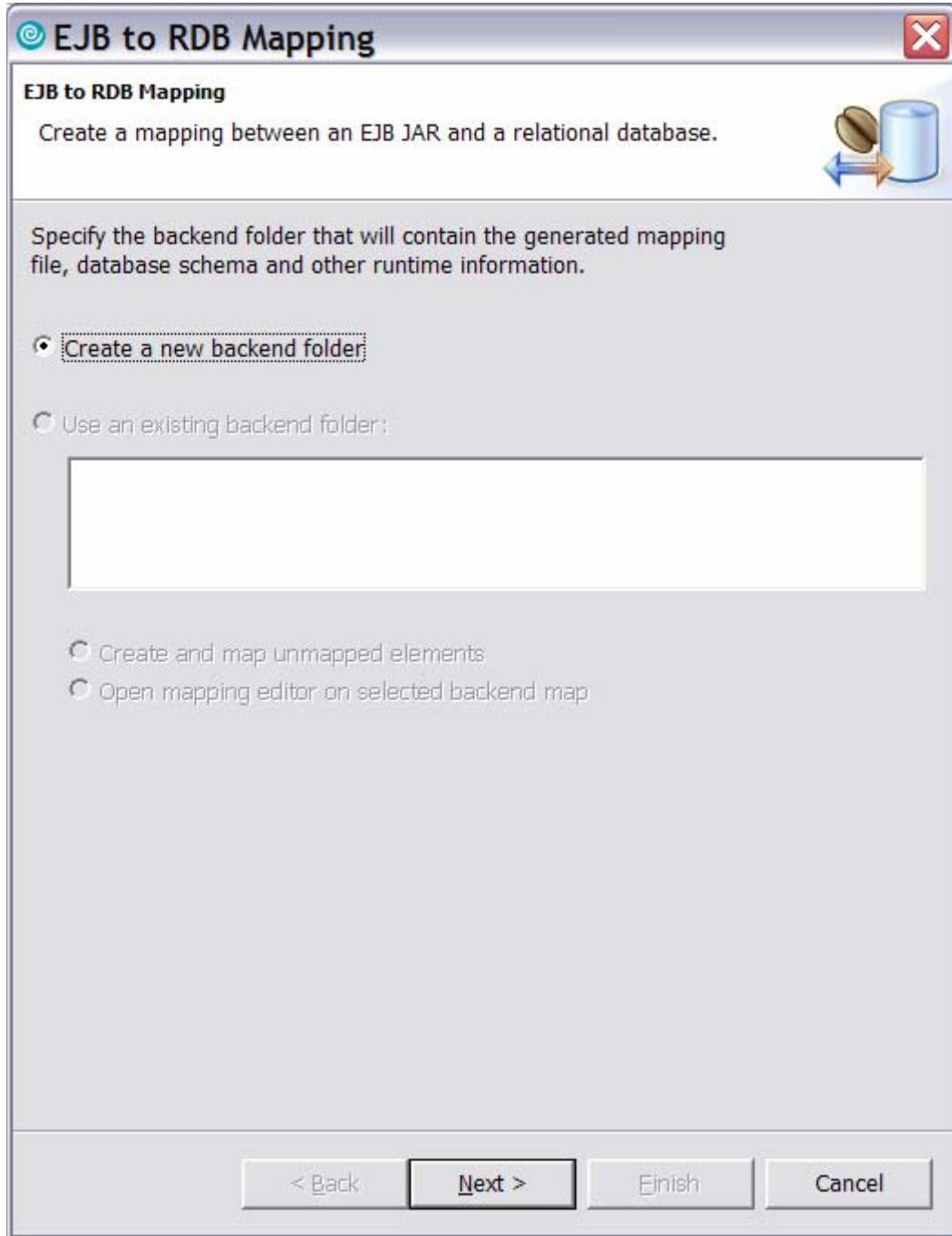


Figure 7: EJB to RDB Mapping wizard

Note: A backend can consist of databases from different vendors or simply alternative mappings and table qualifiers. It is the repository for holding database schemas and mappings of EJB attributes and fields to table columns.

3. Select the type of mapping that you would like to use. This example maps an existing iSeries table, **QCUSTCDT** in library **QIWS**. Select **Bottom-Up** mapping. This means you will map from an existing database to new Java beans. (Figure 8)

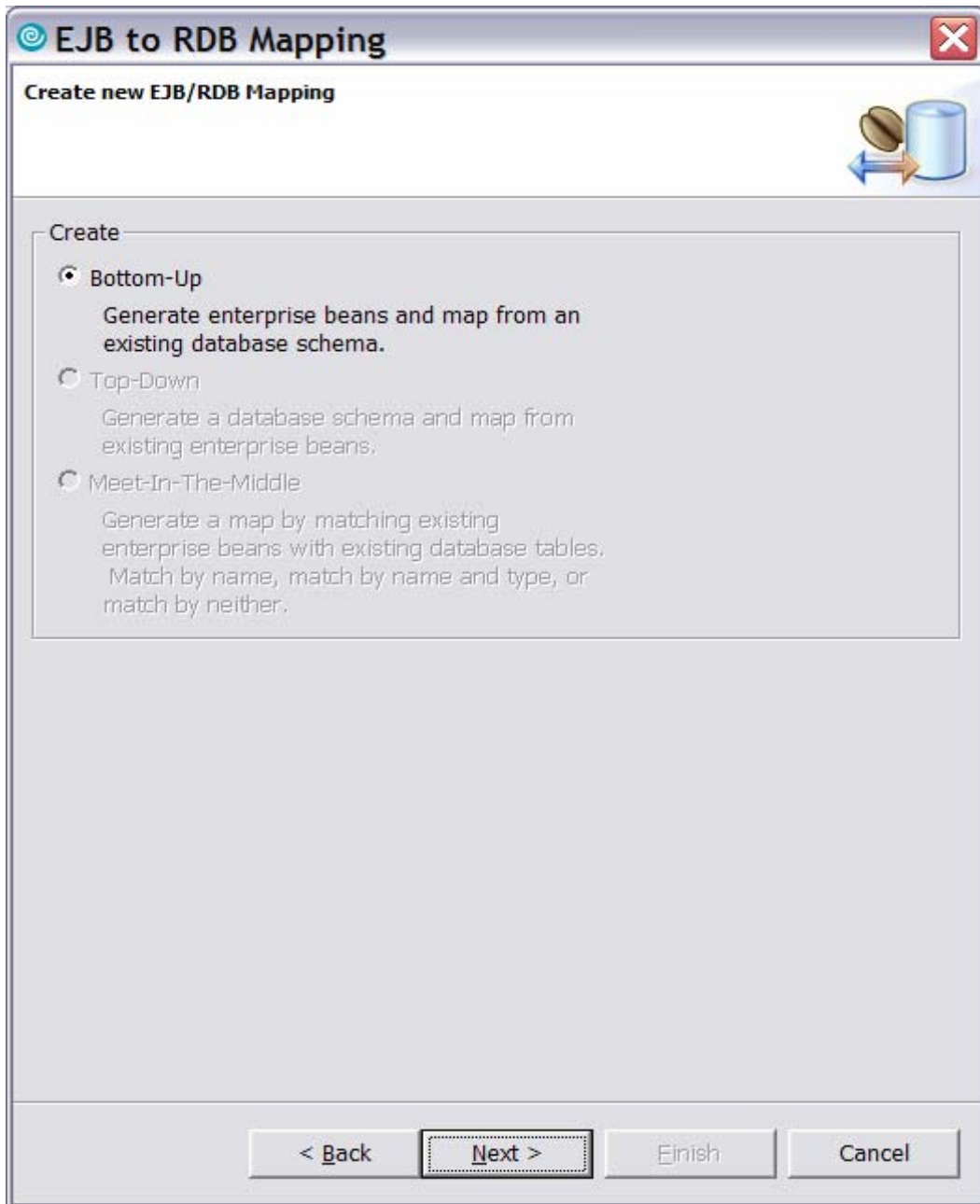


Figure 8: Bottom-Up mapping

Note: **Bottom-Up** is the only allowable selection here. If enterprise beans were already defined, you can generate the back-end database from them or match the enterprise beans to an existing database. You can create new enterprise beans from an existing database, therefore Bottom-Up. (For additional information on the different types of mapping, see Appendix A.)

The process will now ask you to create a new database connection to the database. Once you have created a connection to the iSeries server, it can be reused to create and map other EJBs. The connection is an object stored in Rational Application Developer. You can provide any connection name that you prefer. For a database, use the same name as is set on the i5 server. You can find the database name by using the **WRKRDBDIRE** OS/400 command (see next two screen captures, Figure 9 and 10.) A valid iSeries user ID is required, which allows you to connect to the iSeries database during mapping.

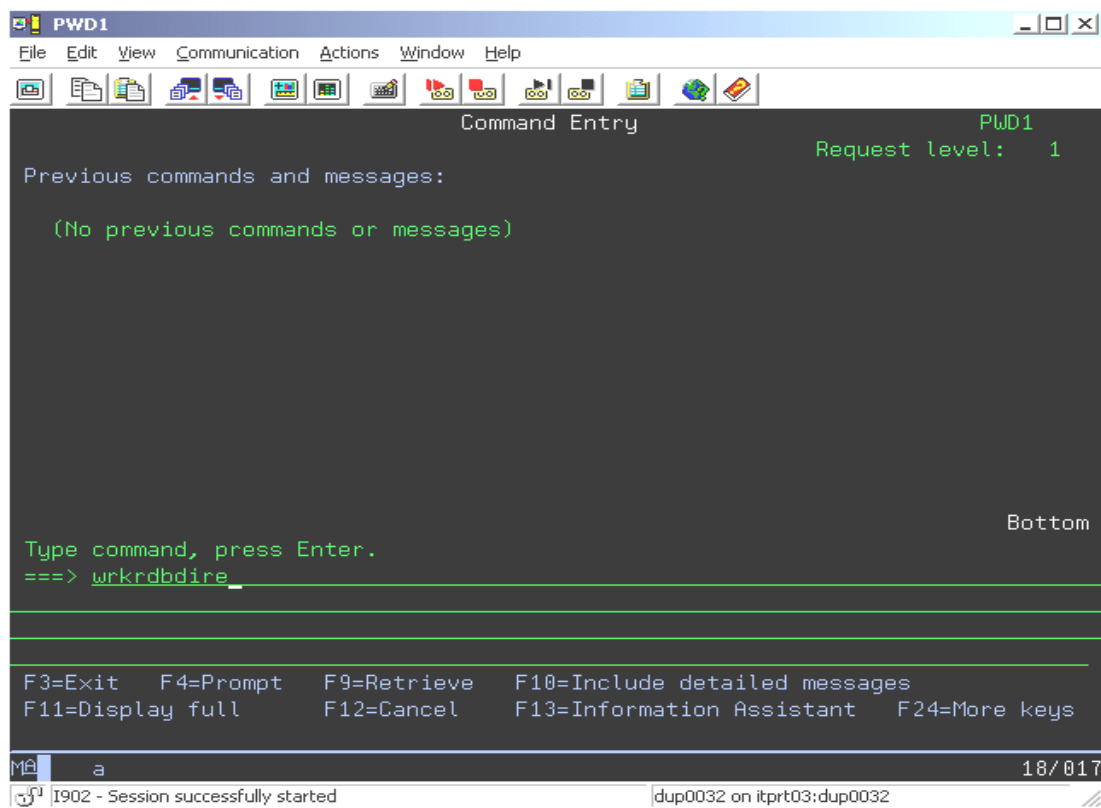


Figure 9: IBM eServer i5 database connection screen 1

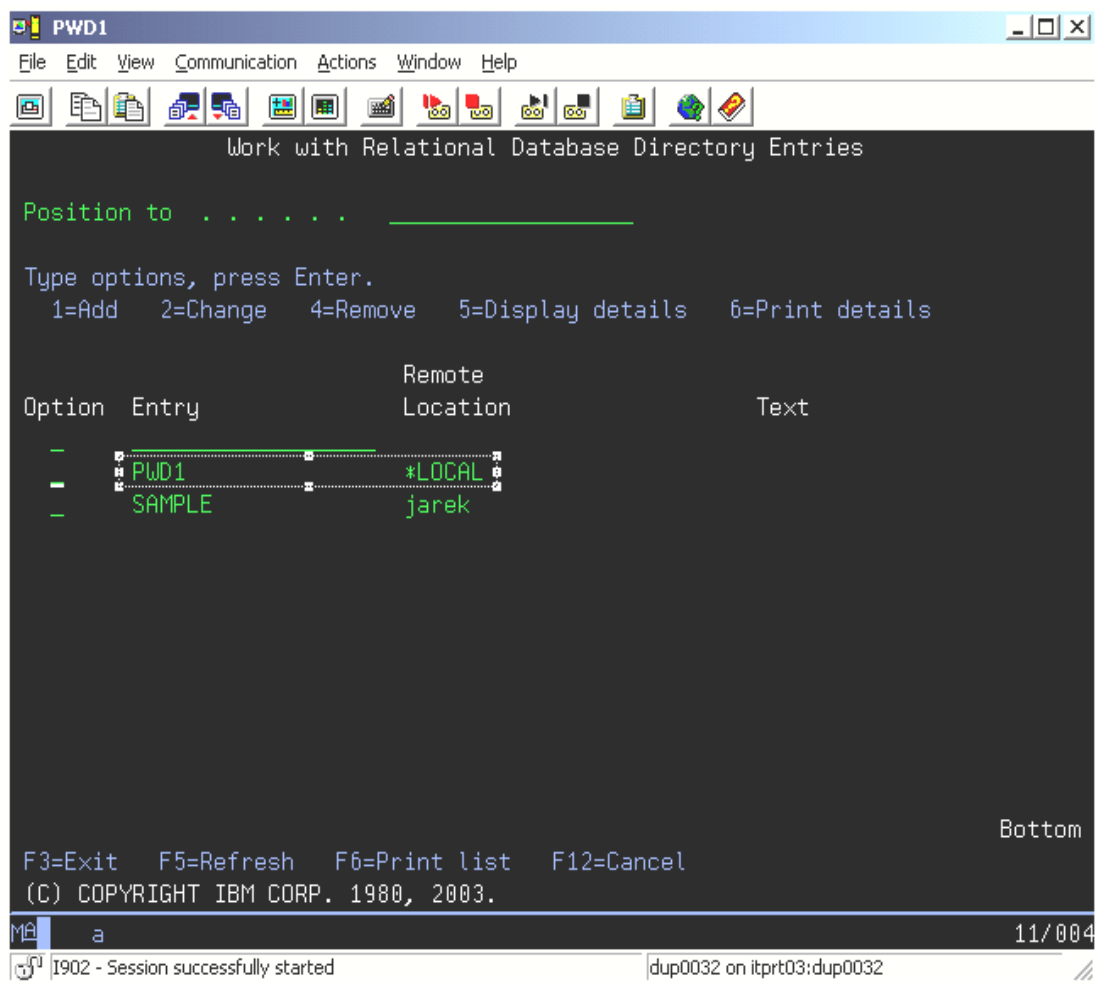


Figure 10: IBM eServer i5 database connection screen 2

4. In the **Database vendor type** field, select **DB2 Universal Database for iSeries V5R3**. The **JDBC driver** field will be completed to indicate **AS/400® Toolbox for Java JDBC Driver**. Fill in the **Host** field to be the TCP/IP host name of the iSeries server you are connecting to. Finally, in the **Class location** field, indicate where to find the jt400.jar file that contains the JDBC driver classes. OS/400 ships with the jt400.jar file in the integrated file system (IFS), in directory **/QIBM/ProdData/HTTP/public/jt400/lib**. Click **Filters**. (Figure 11)

EJB to RDB Mapping

Database Connection
Establish a JDBC connection to a database.

Connection name: iSeriesCon1

Database: SE520B2

User ID: JRUSH

Password: *****

Database vendor type: DB2 Universal Database for iSeries V5R3

JDBC driver: AS/400 Toolbox for Java JDBC Driver

Host: SE520B2

(Optional) Port number:

Server name:

Database Location: Browse...

JDBC driver class: com.ibm.as400.access.AS400JDBCdriver

Class location: K:\QIBM\ProdData\HTTP\Public\jt400\lib Browse...

Connection URL: jdbc:as400://SE520B2

Filters...

Use existing connection:
Existing connection:

< Back Next > Finish Cancel

Figure 11: Database connection wizard

5. Connection filters are next. The default for the iSeries connection is shown in the figure that follows (Figure 12). It indicates that tables in all iSeries libraries *except* those libraries that have **QSYS**, **\$**, or **#** as part of their name are included in the mapping wizard. For performance reasons, you will want to provide another filter. Click **Add Filter**.

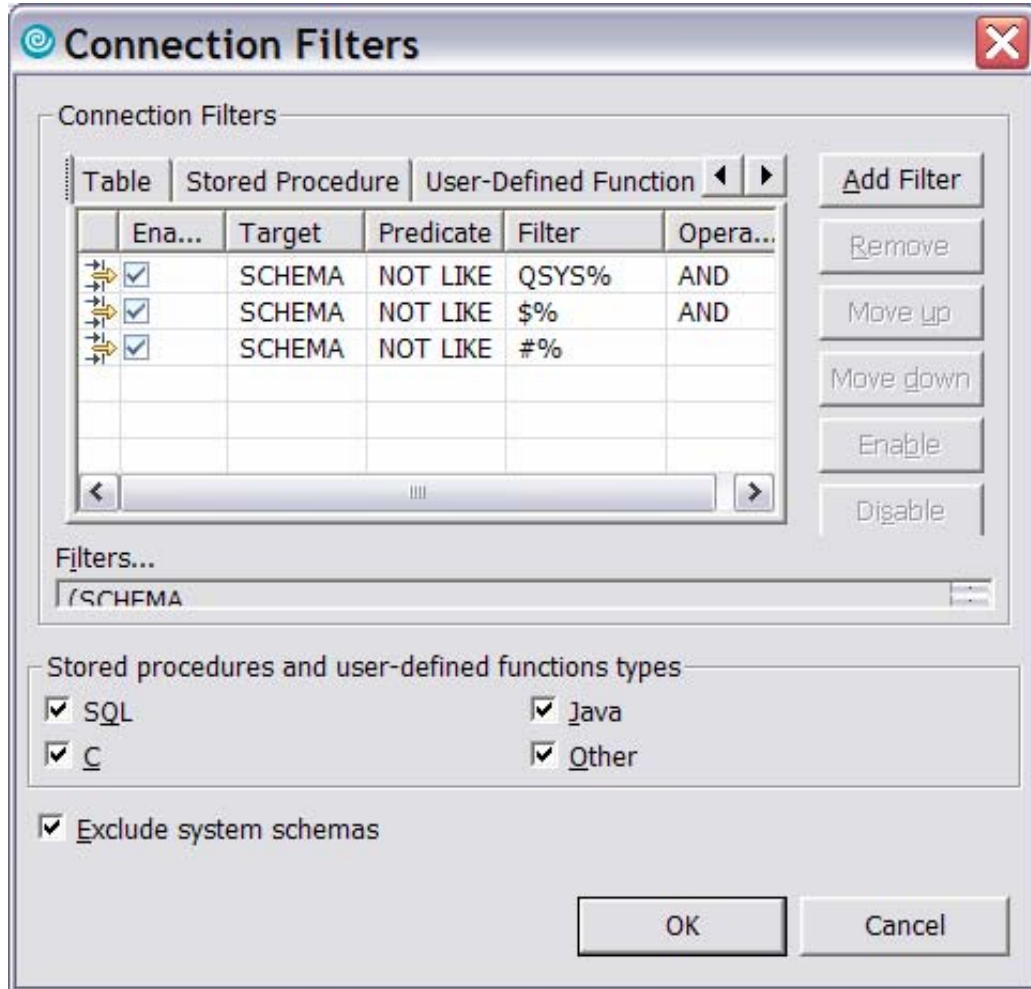


Figure 12: Add filters

5. In the *Add Filter* dialog, use the following values: For the **Select** field choose **SCHEMA** from the pull down menu; and for the **where name** field choose **LIKE** and type in **QIWS**. This will reduce the number of tables you will see in the mapping wizard to only those in library **QIWS**. Click **OK**. (Figure 13)

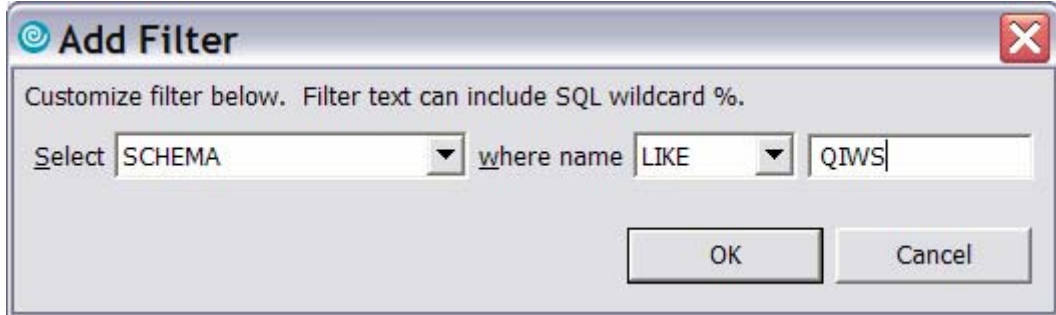


Figure 13: Add Filter dialog

6. The finished filter dialog will look like the following screen. Click **OK**. (Figure 14)

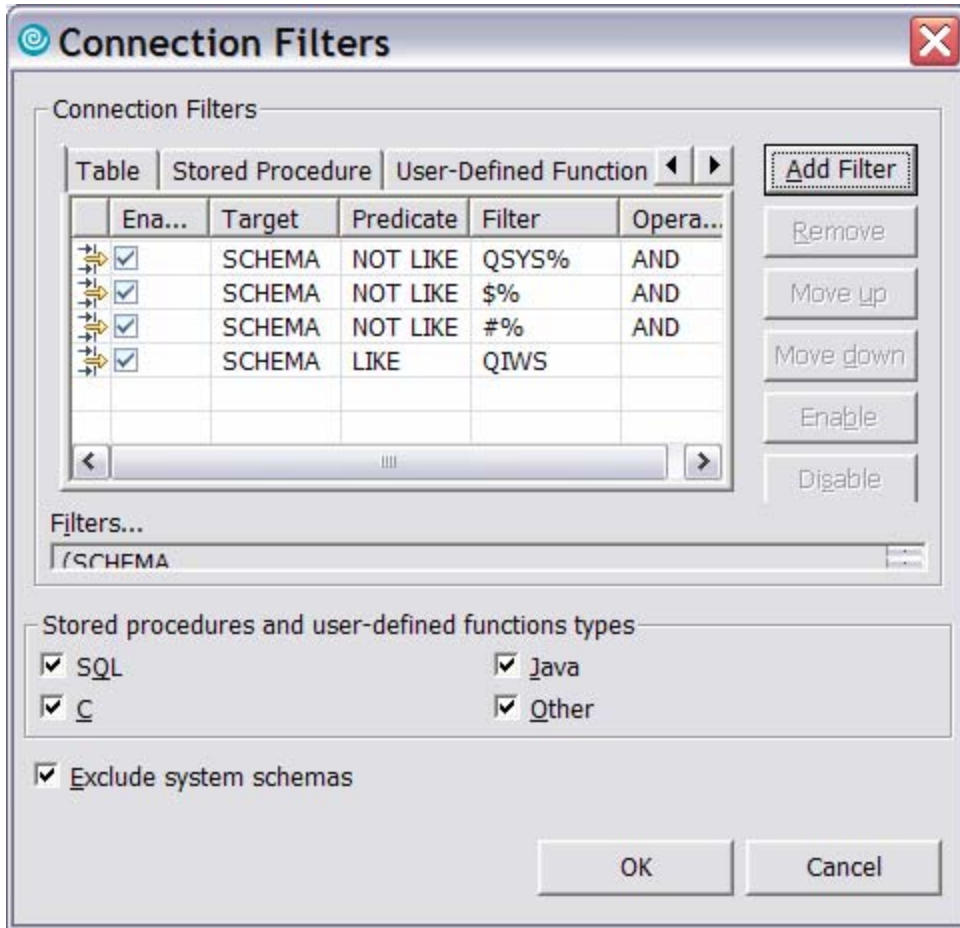


Figure 14: Finished filter dialog

7. Back in the wizard, click **Next**. The wizard will connect to the iSeries server through the connection and filter just created. A list of tables is shown that are available to map to the new EJBs.
8. Select the table that you would like to map to, which in this case is **QCUSTCDT**. Click **Next**. (Figure 15)

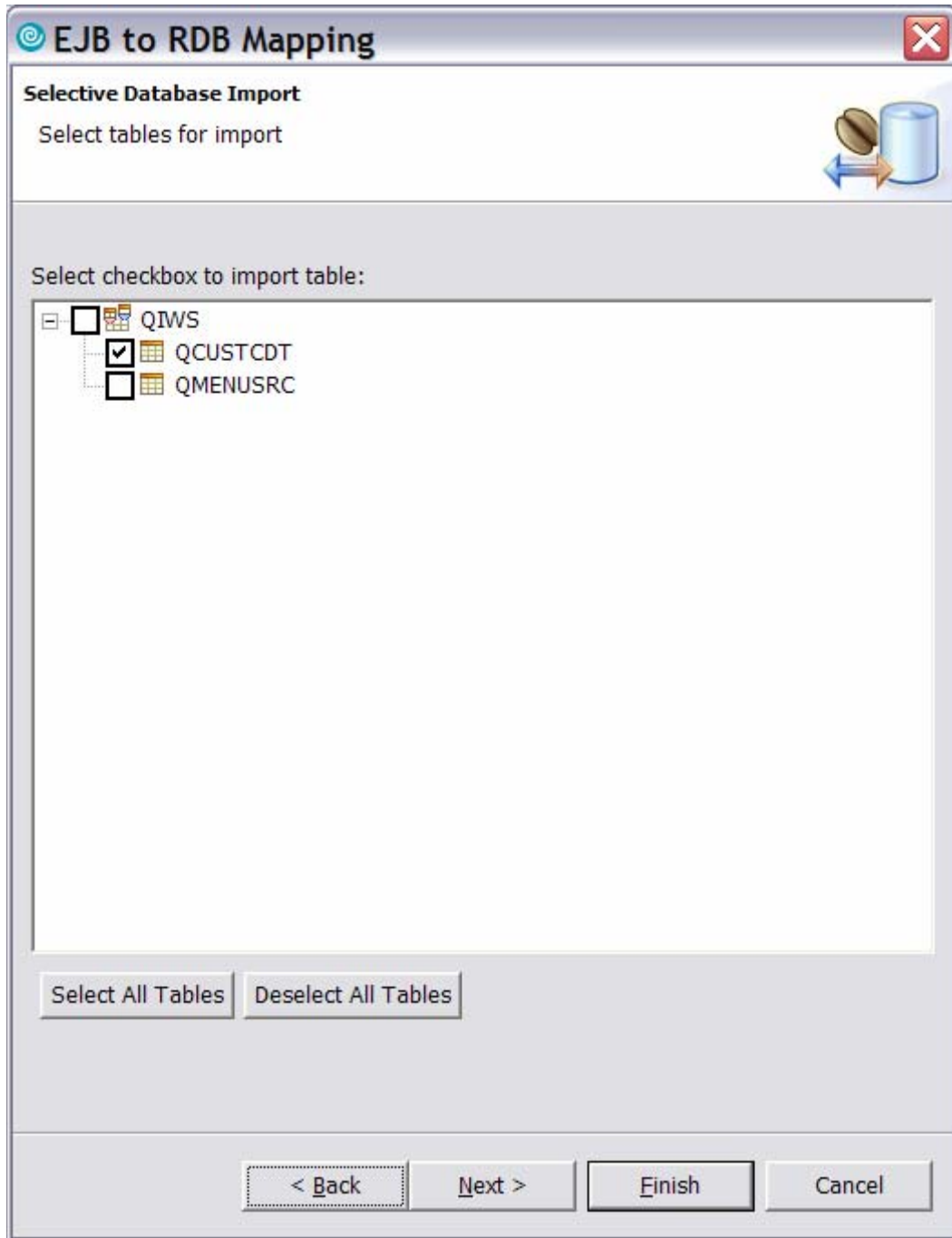


Figure 15: EJB to RDB Mapping wizard, Selective Database Import

9. On the *EJB to RDB Mapping* screen, under **CMP Version** select **2.x**. In this example, a prefix for the generated EJBs is not specified. Instead the package name **se520b2** is used that matches the iSeries server you are connected to.
10. Click **Finish**. (Figure 16)

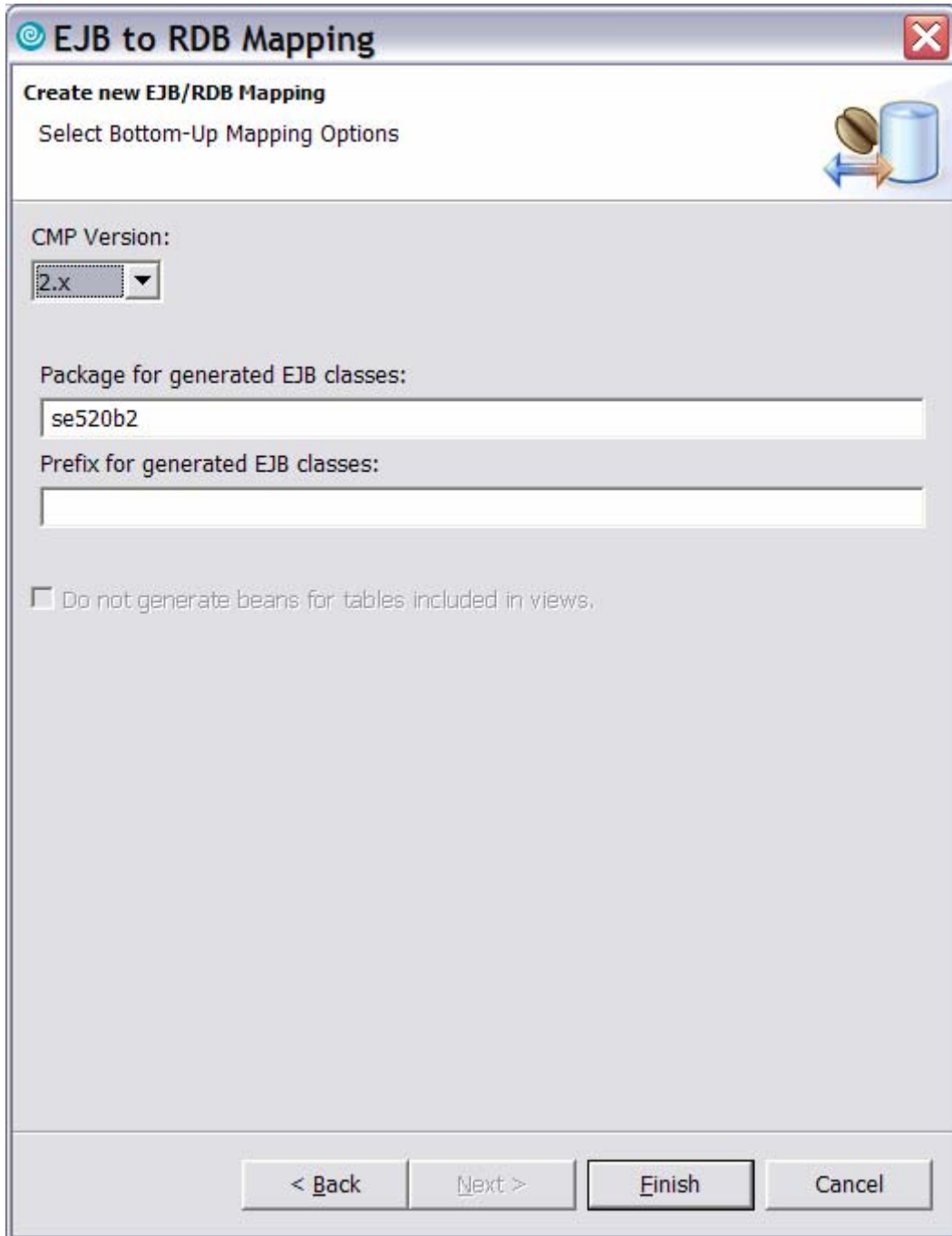


Figure 16: Create new EJB/RBD Mapping

11. The wizard will take you to the mapping editor as seen below. (Figure 17)

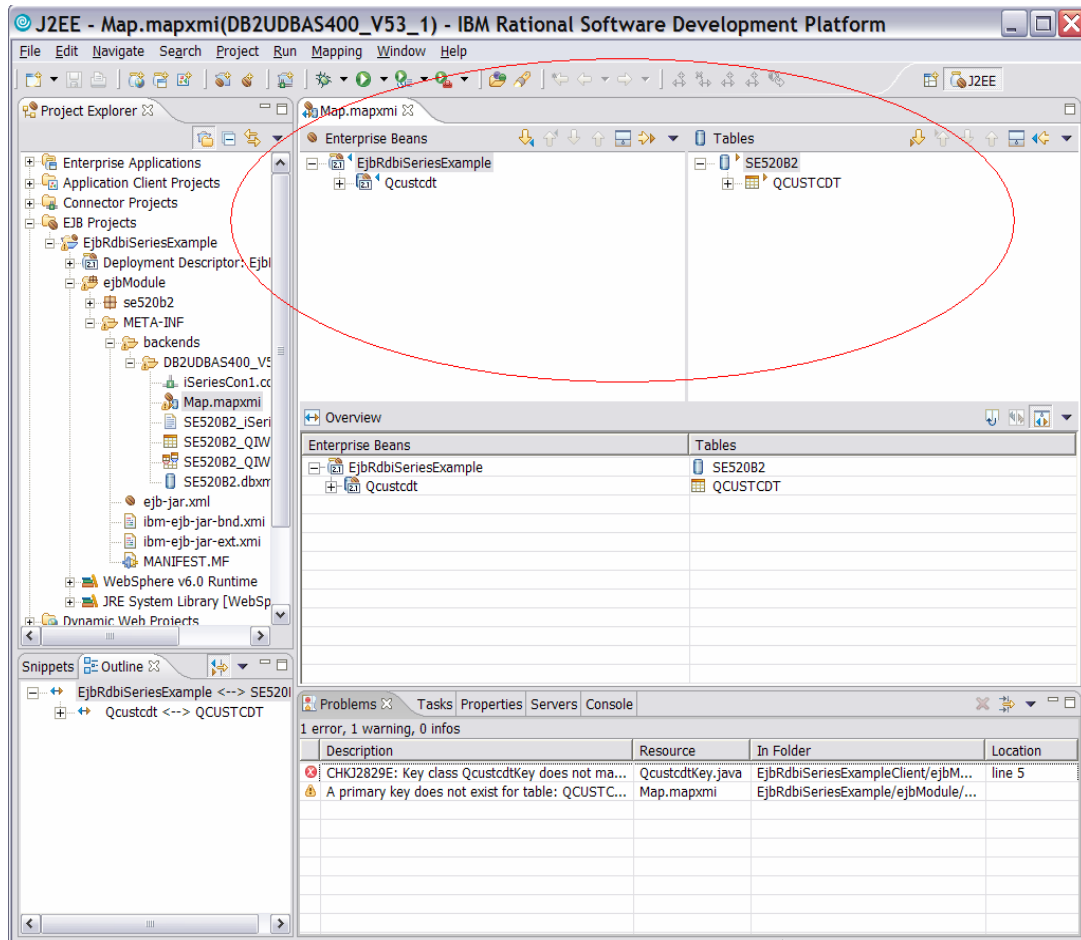


Figure 17: Mapping editor

In the circled area, see the generated EJBs and the iSeries connection and table. Note the errors in the *Tasks* pane in the *Problems* tab. The messages indicate that you have to map and identify a primary key that will be used to instantiate the entity EJB. To remedy these issues, map and identify which table column will be used and mapped to the entity bean attribute as the primary key.

12. Right click **QCUSTCDT** and then click **Open Table Editor**. (Figure 18)

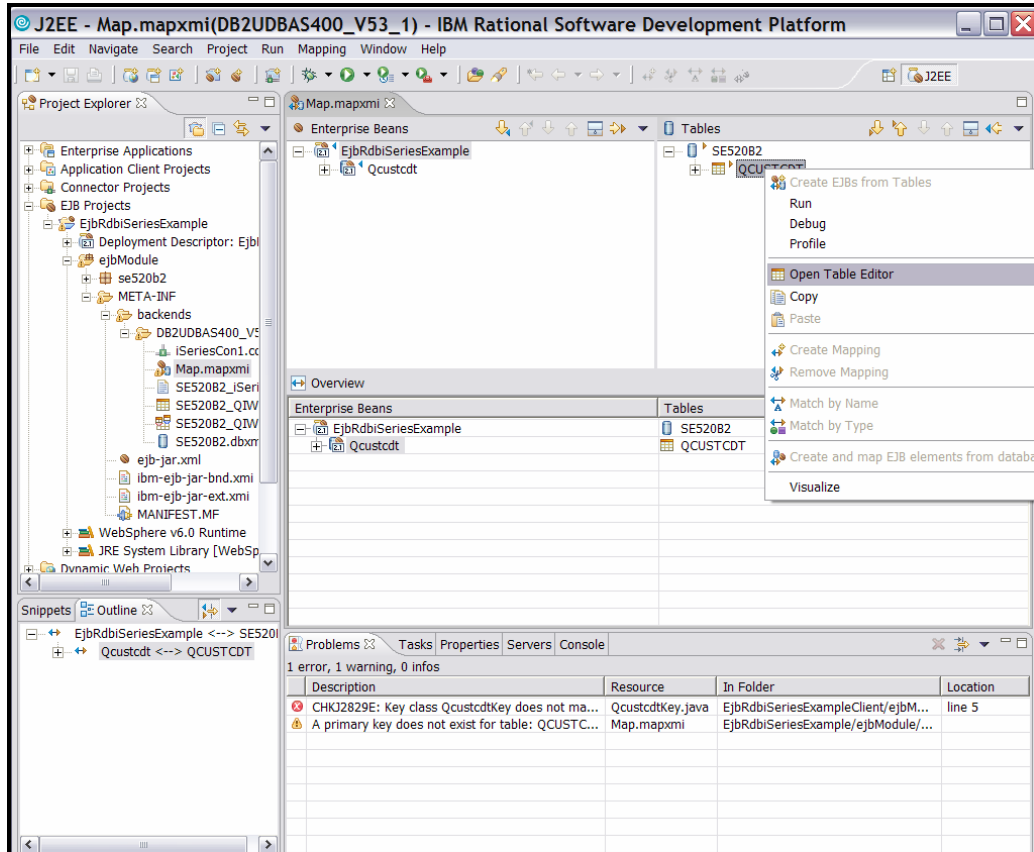


Figure 18: Open Table Editor

- When the table editor opens, select the *Primary Key* tab. For this example, select **CUSNUM:NUMERIC** in the left hand **Key column(s)** pane and click > to move it over to the **Column(s) in Primary Key** pane. (Figure 19 in red.)

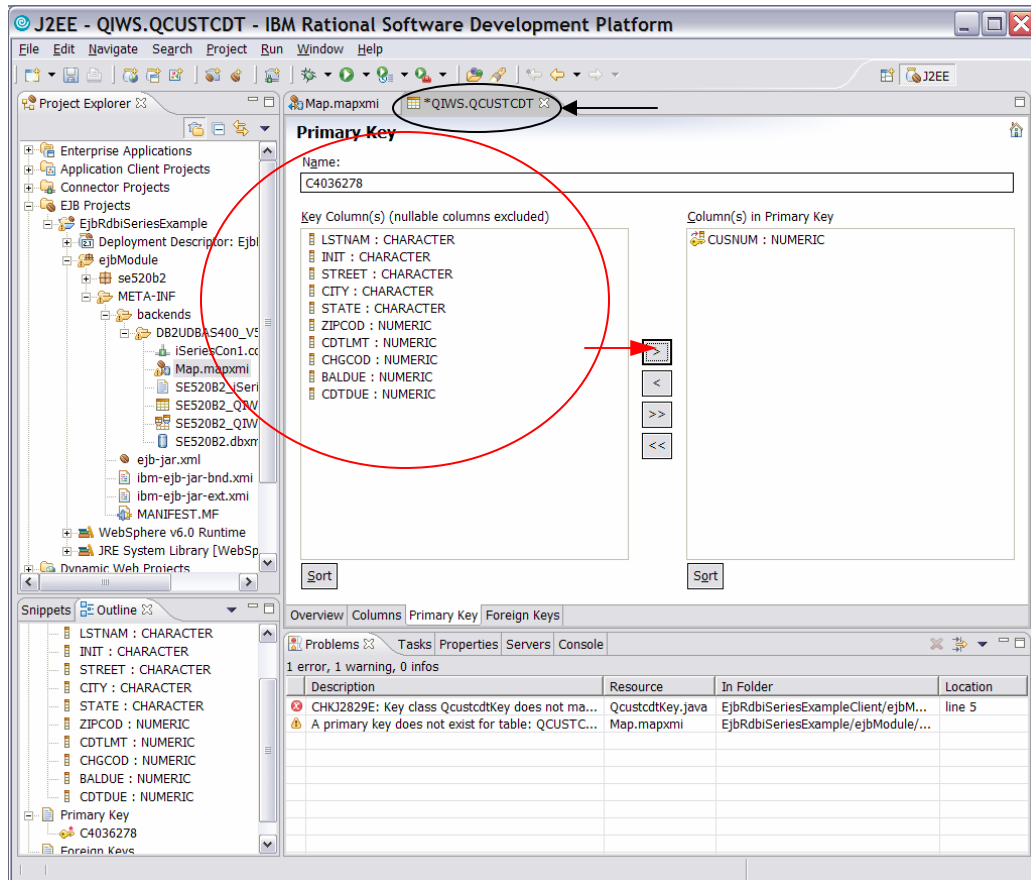


Figure 19: Primary Key

- Close the table editor by clicking on the **X** on the **QIWS.QCUSTCDT** tab and click **Yes** to save changes. (Figure 19 in black.)

15. Right click **Qcustcdt** and select **Open Deployment Descriptor Editor**. (Figure 20)

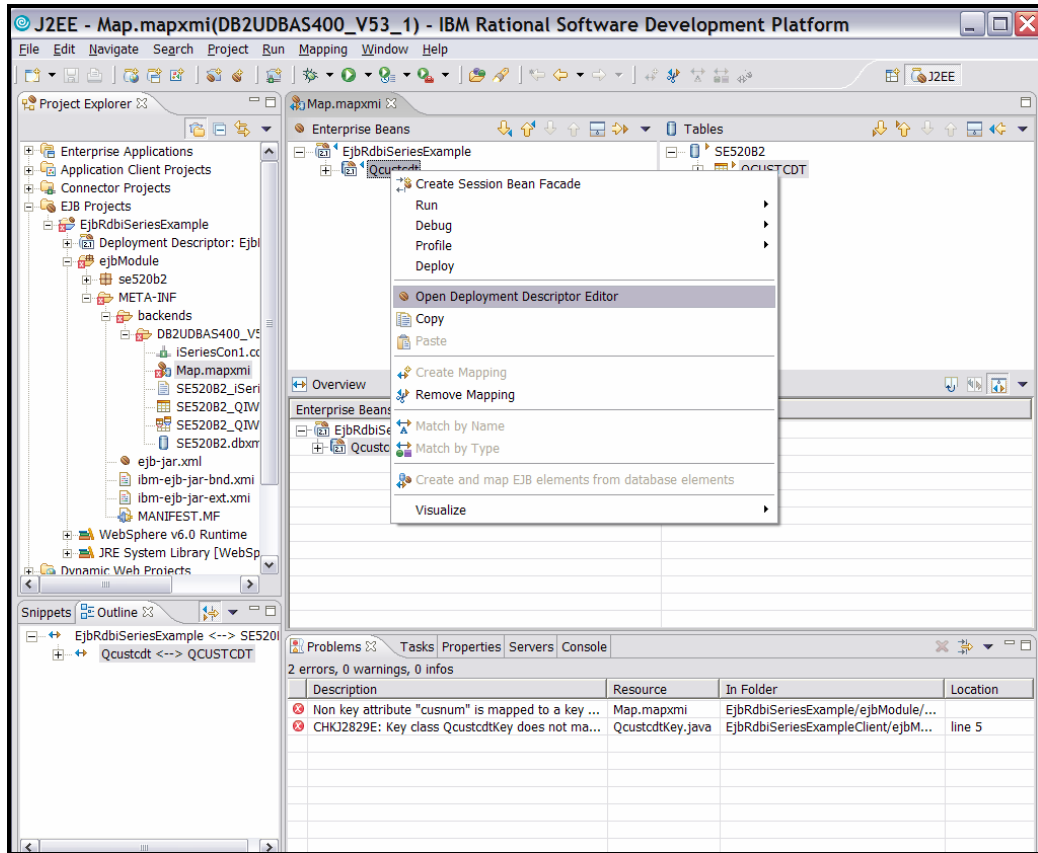


Figure 20: Open Deployment Descriptor Editor

- On the *Bean* tab in the *Deployment Descriptor Editor* screen, select **cusnum** in the **CMP Fields** pane and click **Add to Key**. (Figure 21 in red)

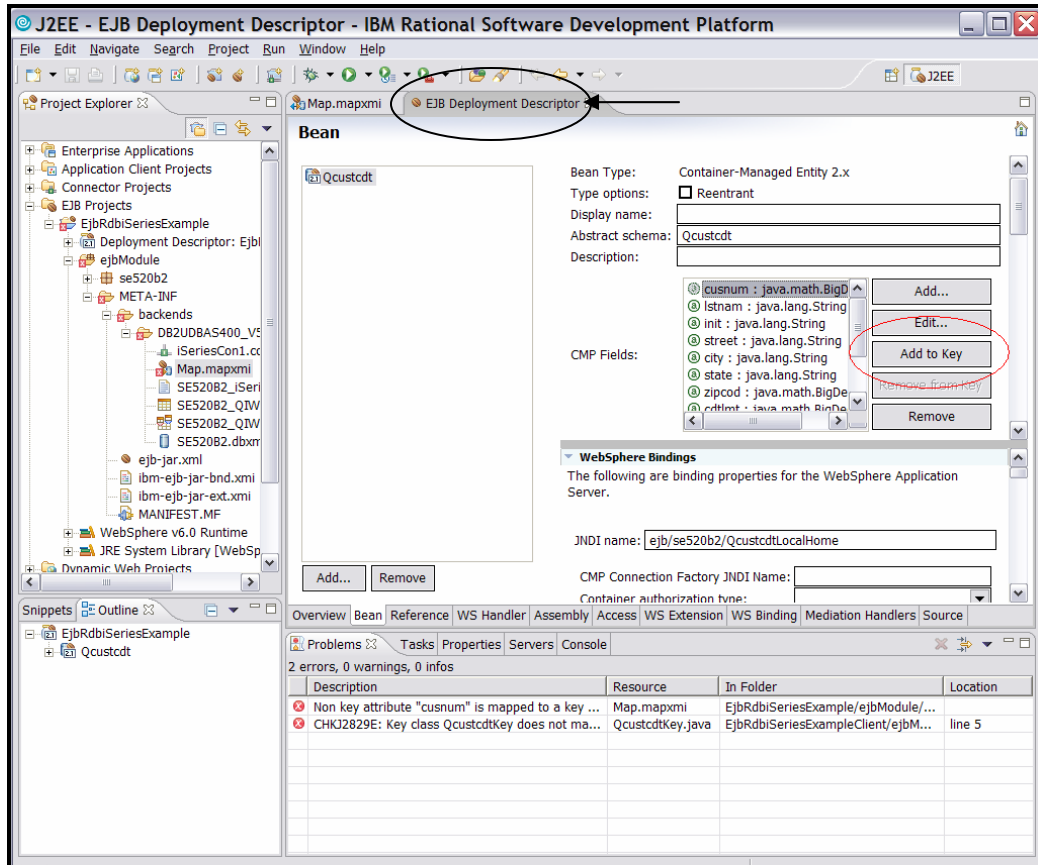


Figure 21: Add to Key

- Close the *Deployment Descriptor Editor* by clicking the **X** on the **EJB Deployment Descriptor** tab (Figure 21 in black) and click **Yes** to save changes.

18. The errors listed in the *Tasks* pane will go away and the mapping editor will look as follows (Figure 22) after expanding **QCUSTCDT** in both the **Enterprise Beans** and **Tables** panes, with **cusnum** in the EJB mapped as the key to the **CUSNUM** column in the **QCUSTCDT** table. Close the **Map.mapxmi** tab by clicking on the **X** (in black).

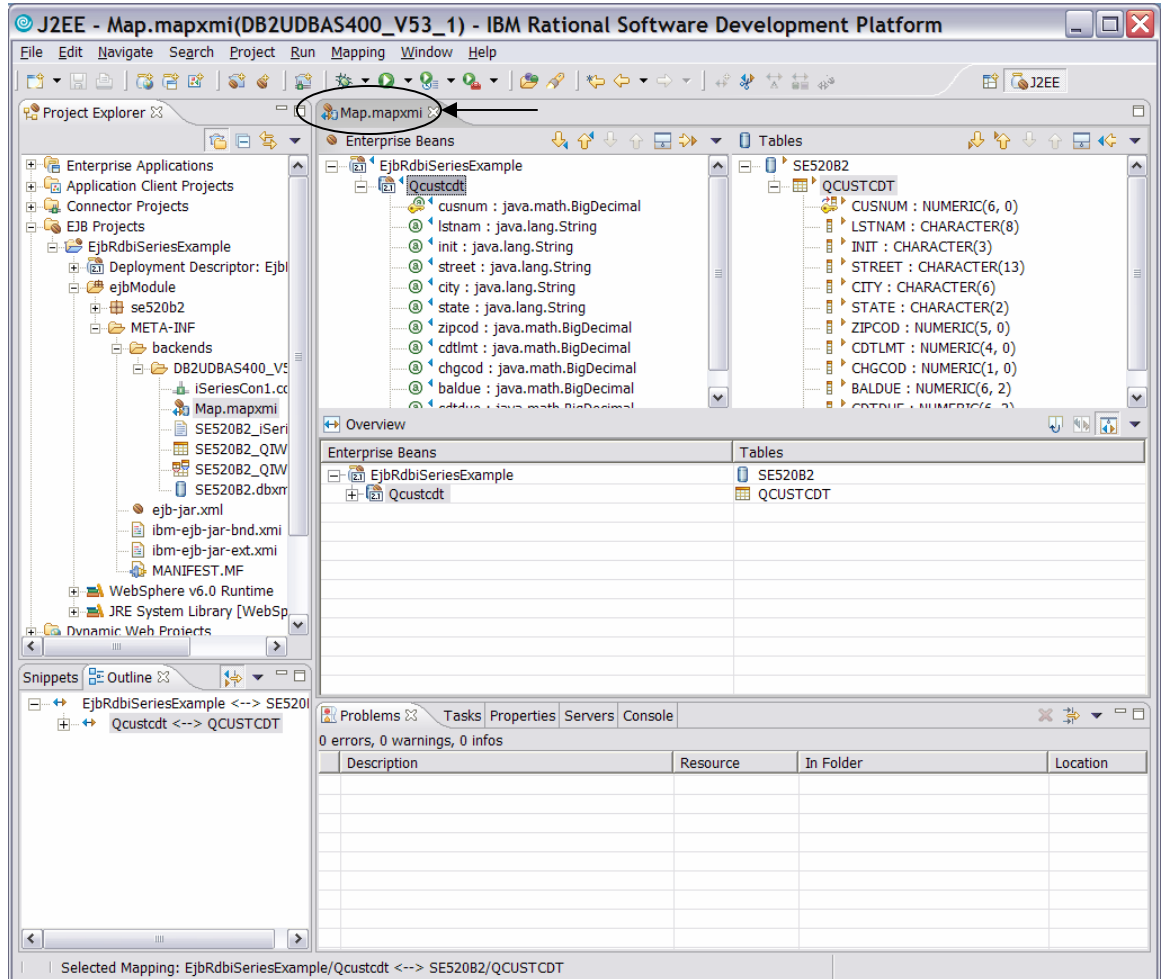


Figure 22: Mapping beans to tables

Specify a default data source for the EJBs to use

Now you will specify a default data source for the EJBs to use:

1. Right click on **EJB Deployment Descriptor** and select **Open**.

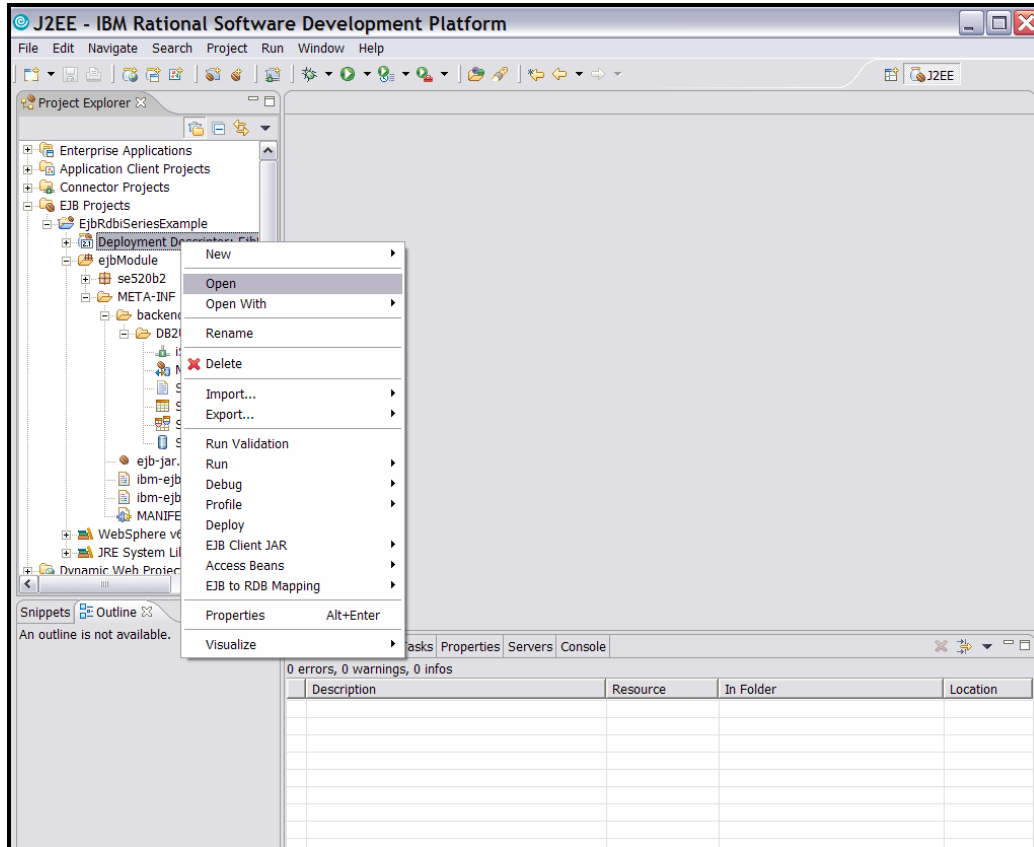


Figure 23: Open EJB Deployment Descriptor

2. On the right side of the *Overview* tab, update *JNDI – CMP Connection Factory Binding* values. Enter **jdbc/iSeriesToolbox** in the *JNDI name* field and **Per_Connection_Factory** in the *Container authorization type* field (Figure 24 in red).

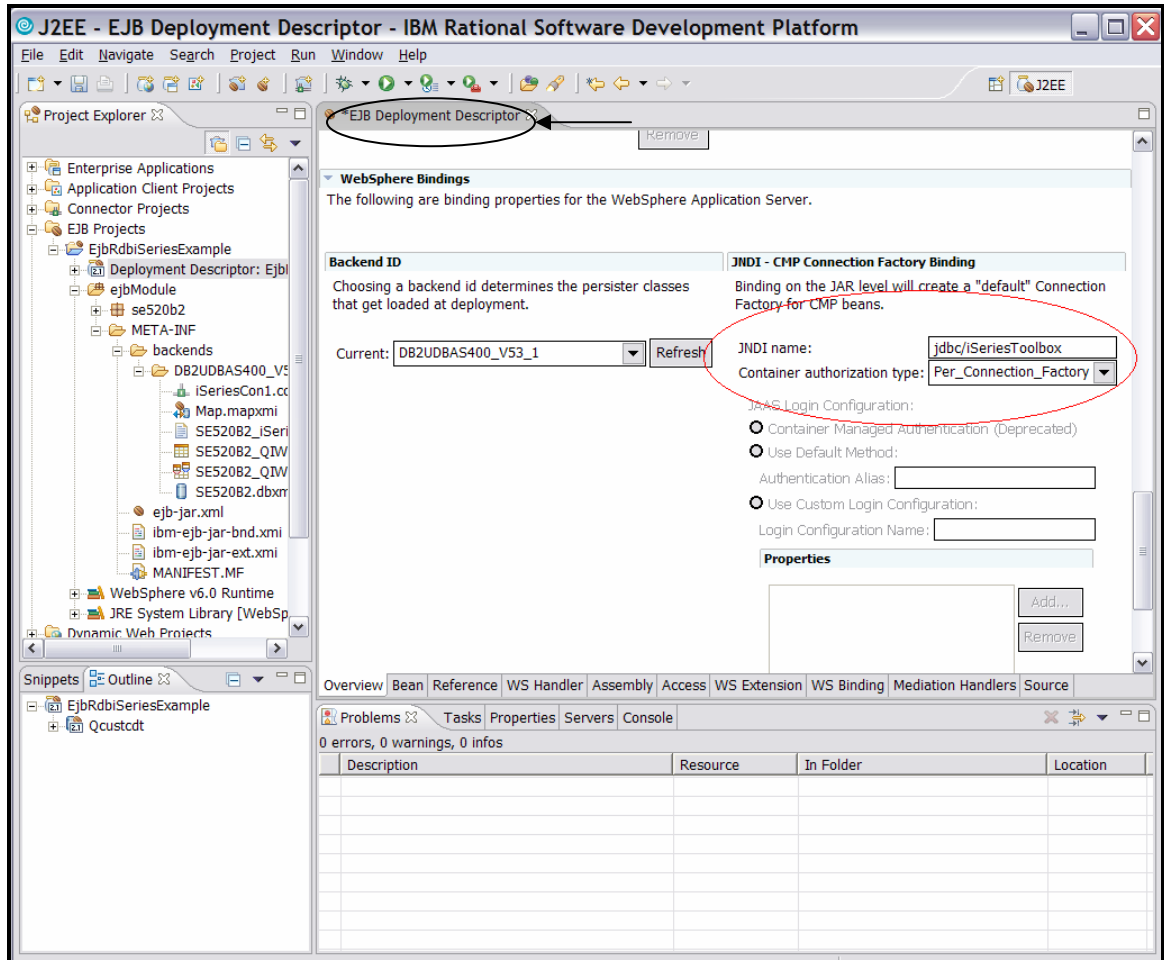


Figure 24: Update JNDI-CMP Connection Factory Binding values

3. Close by clicking the **X** in the *EJB Deployment Descriptor* tab (Figure 24 in black) and click **Yes** to save the changes.

4. Right click **EjbRdbiseriesExample** project and select **Deploy** to generate RMIC code for deployment of your beans. (Figure 25 and 26)

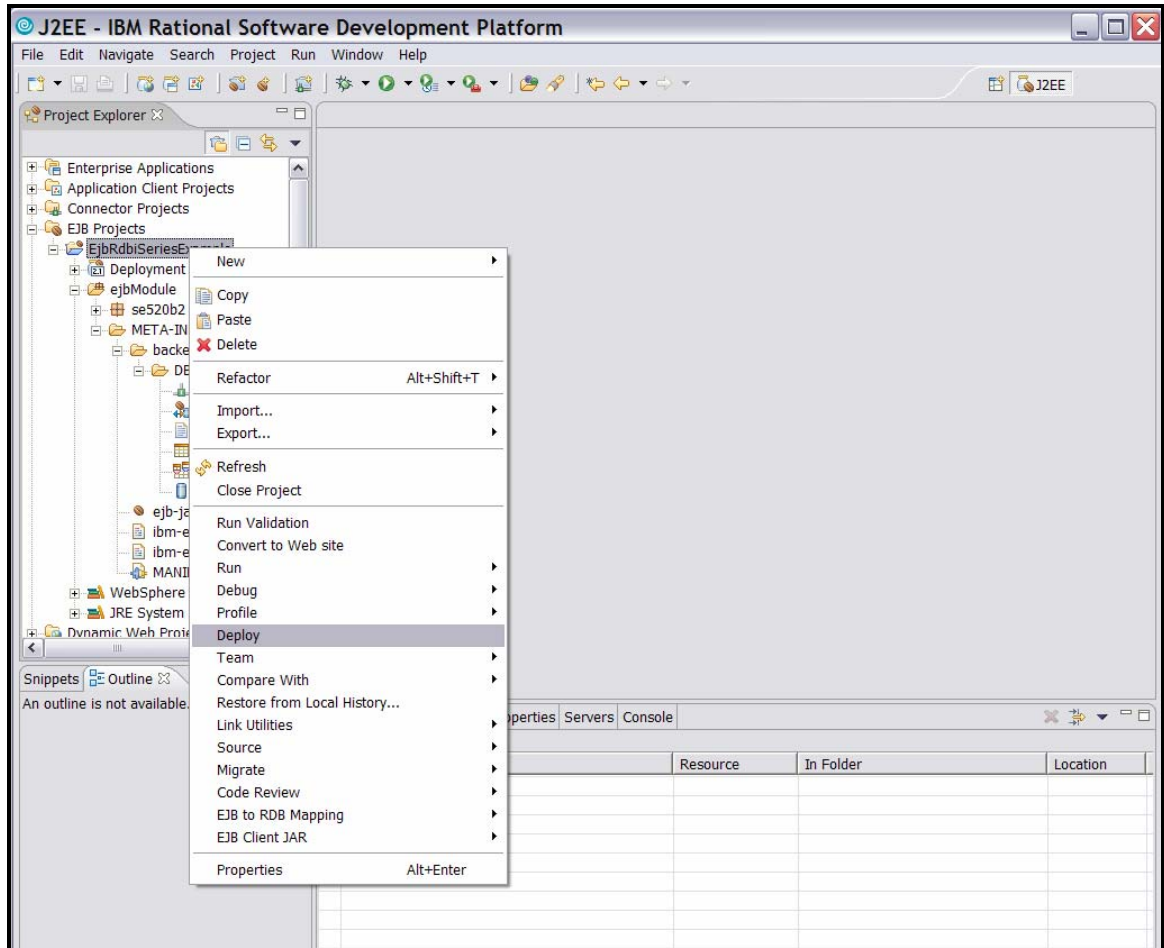


Figure 25: Deploy the EJB

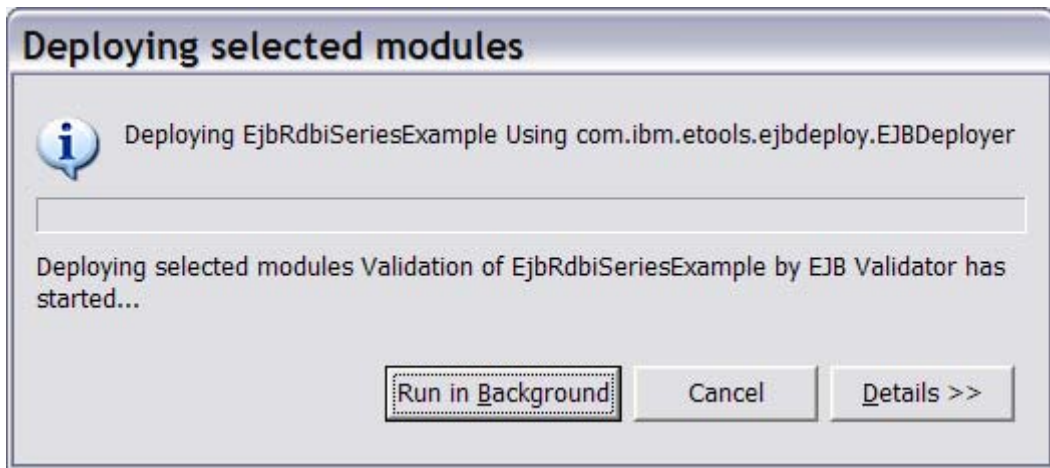


Figure 26: Deploying selected modules

Create a WebSphere test server environment

The Rational Application Developer IDE ships with several WebSphere integrated runtimes that might be used for testing the components and artifacts right in the tool itself. You can create Version 5, Version 5.1, or Version 6.0 servers and those servers can be configured in Express (does not contain all J2EE functions for example, EJB support) or Base (fully J2EE compliant) versions. In this example, we will create a base 6.0 test server environment.

1. First, create a new server and server configuration by clicking **File > New > Other...** (Figure 27)

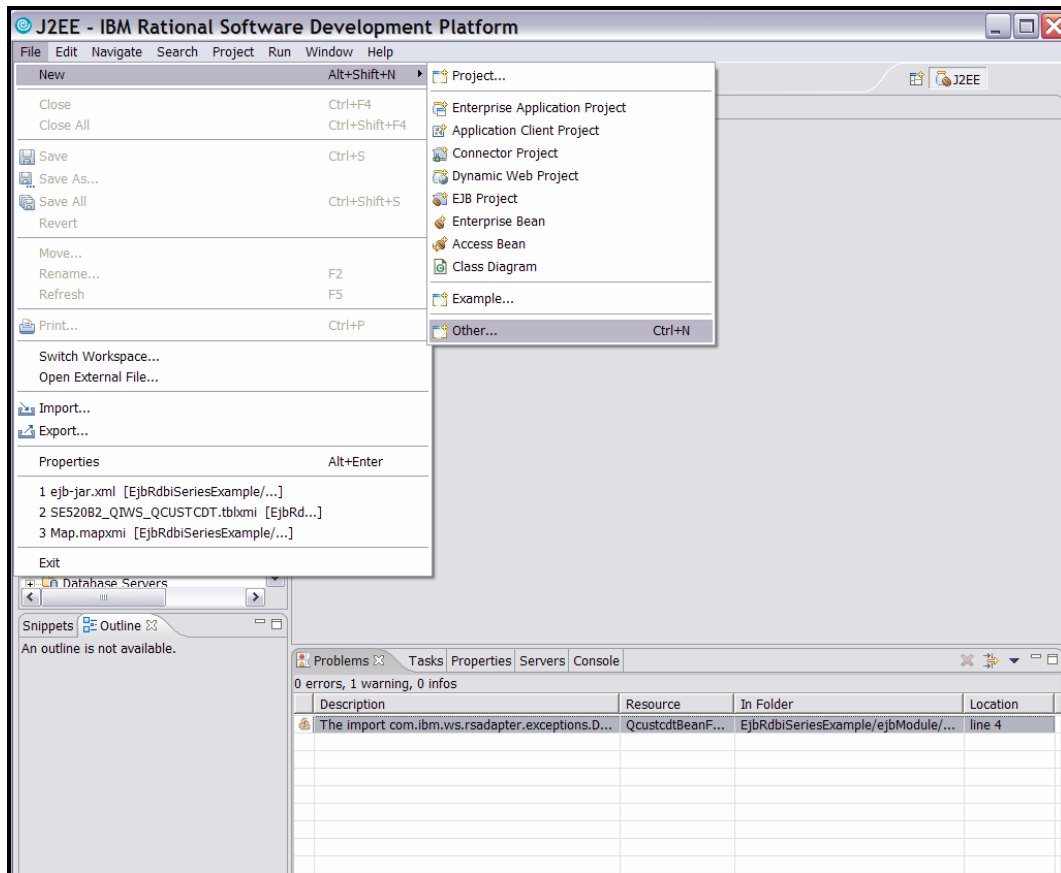


Figure 27: Create a new WebSphere runtime environment

2. On the *Select a wizard* screen, click **Next**. (Figure 28)

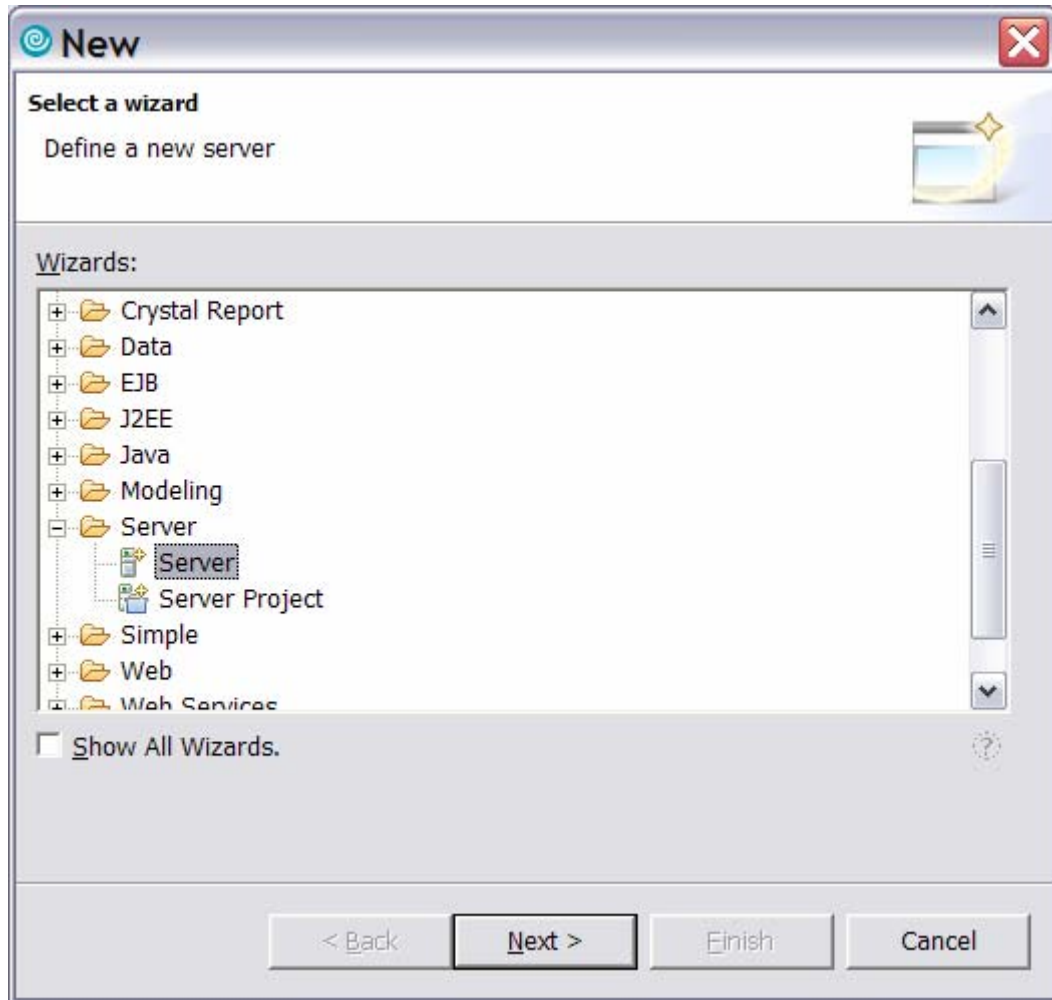


Figure 28: Select a wizard, Define a new server

3. Let the **Host name** default to **localhost** and select **WebSphere v6.0 Server**. Click **Next**. (Figure 29)

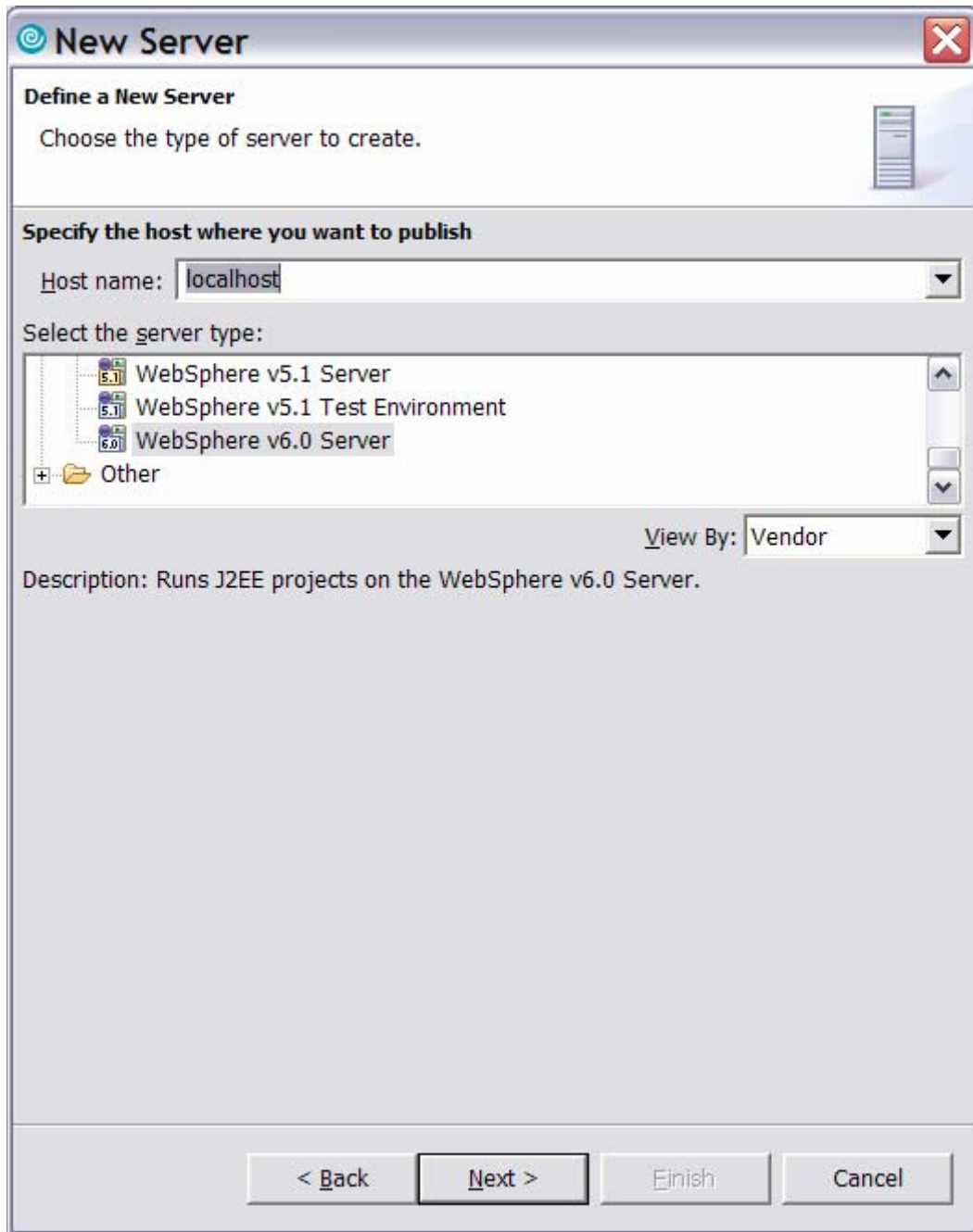


Figure 29: New Server wizard: Choose the type of server to create.

4. Accept the defaults and click **Next**. (Figure 30)

New Server

WebSphere Server Settings

Input settings for the new WebSphere server.

WebSphere profile name: default

Server admin port number (SOAP connector port): 8880

Server name: server1

Run server with resources within the workspace

Server type:

BASE or Express server

Network deployment server

Network deployment server name:

The server name is in the form of:
<cell name>/<node name>/<server name>
For example, localhost/localhost/server1.

Click this button to detect the server type.

Enable security

Current active authentication settings:

User ID:

Password:

< Back Next > Finish Cancel

Figure 30: New server wizard. WebSphere Server Settings screen.

5. Select **EjbRdbiSeriesExampleEAR** and click **Add** (Figure 31). Then click **Finish** (Figure 32).

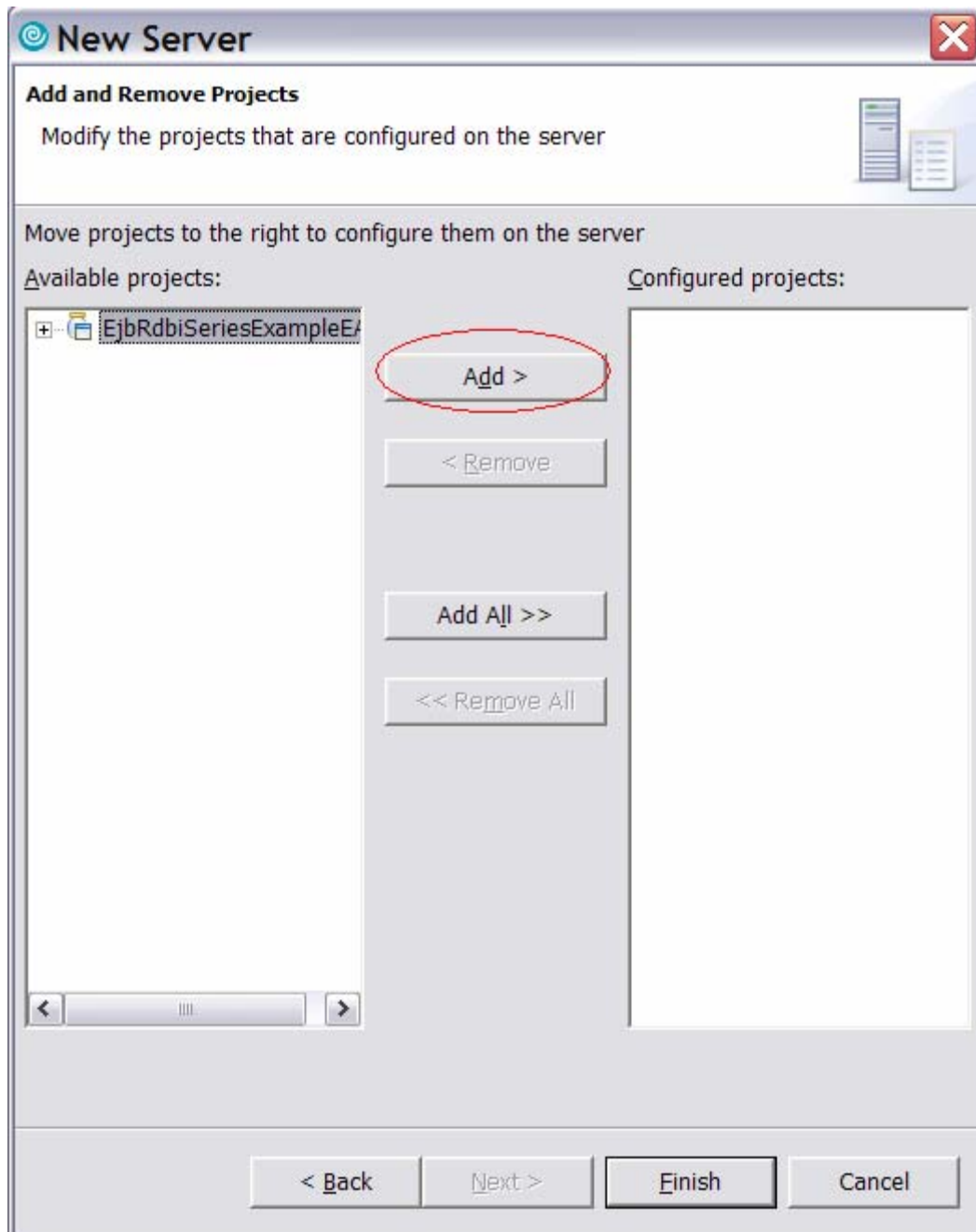


Figure 31: Add new server



Figure 32: New Server wizard Finish

Congratulations, you now have a WebSphere test environment. Note that the WebSphere 6 server might now try to start.

Modify the server configuration for the testing

Now that you have a WebSphere test environment, you need to modify the server configuration for the testing. You will access the WebSphere 6 administrative console and edit the server configuration to add a data source for the iSeries server. The database will reside in the WebSphere server EJB container and EJBs will operate on the data. The following steps show how to open the Administrative console.

1. Make sure your WebSphere V6.0 test server is started. Under the status column you might see **Started** or **Debugging**. If the status indicates **Stopped**, right mouse click on your WebSphere V6.0 test server and select **Start**. (Figure 33)

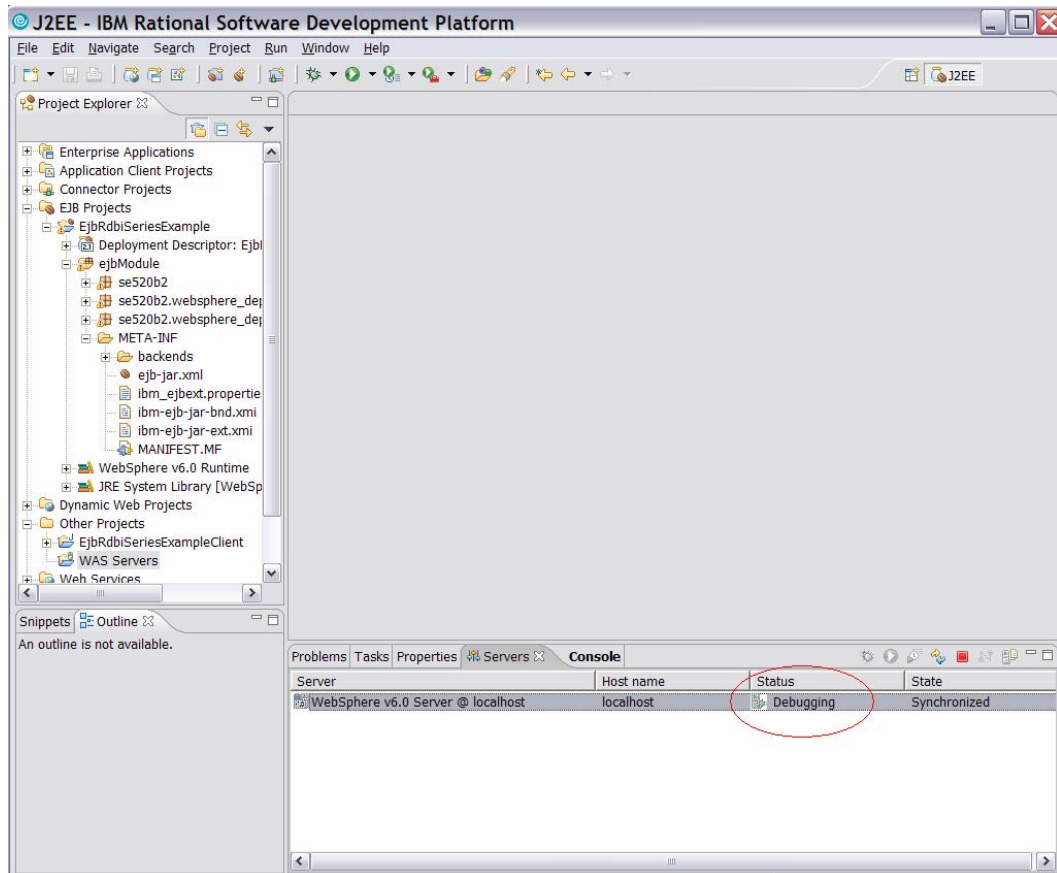


Figure 33: WebSphere V6.0 @ localhost started and Debugging started

2. Right click the **WebSphere V6.0** test server and select **Run administrative console** (Figure 34 circled in red).

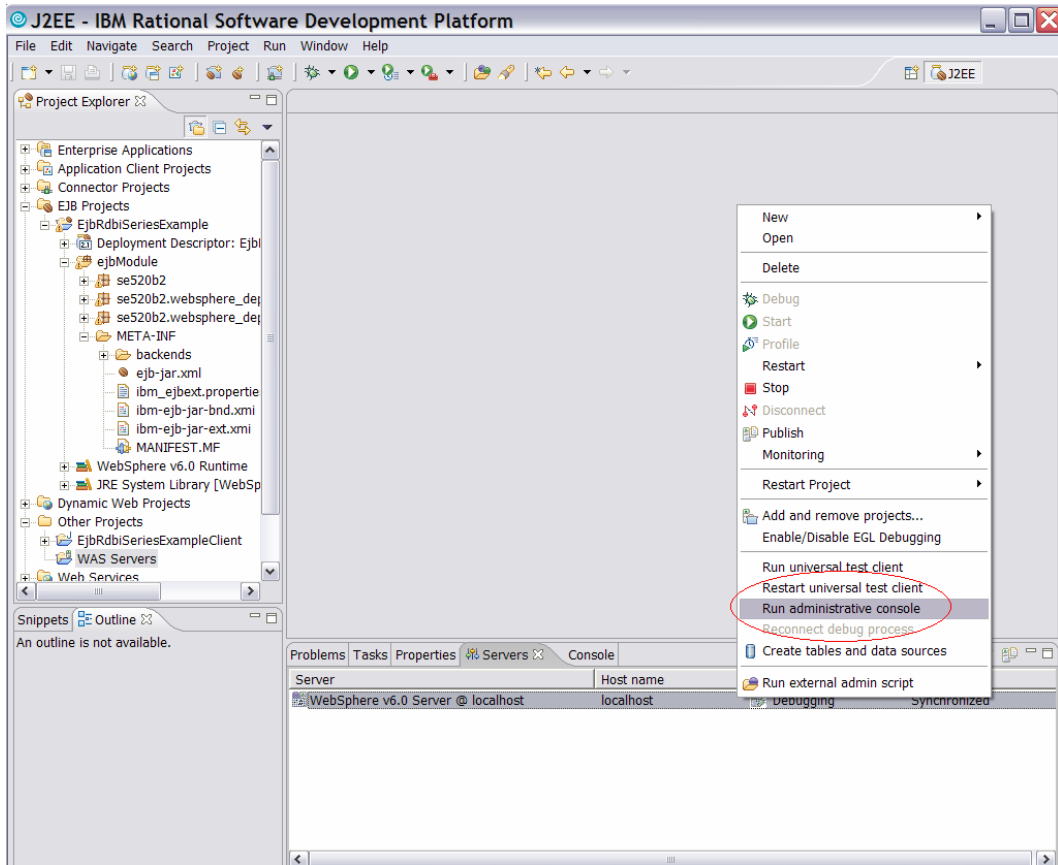


Figure 34: Run administrative console

3. Sign on as the administrator in the **User ID:** field and click on **Log in.** (Figure 35)

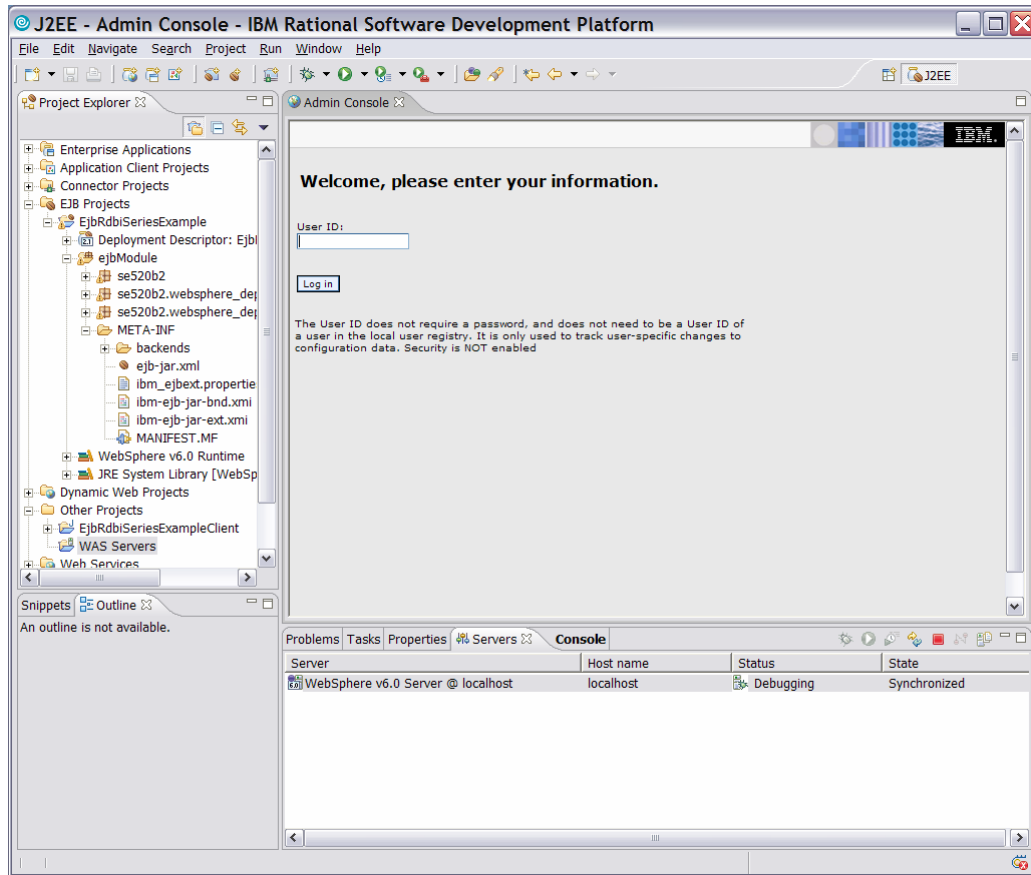


Figure 35: Login screen

4. The following welcome panel appears in the **Admin Console**. (Figure 36)

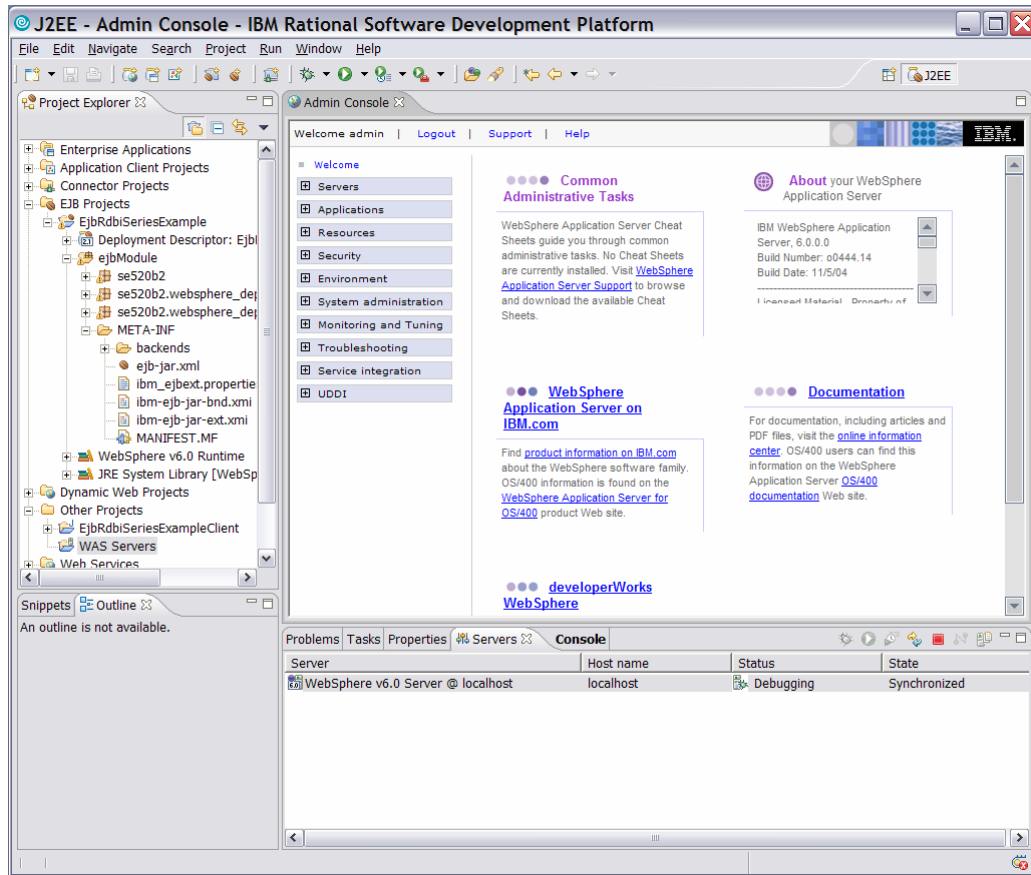


Figure 36: Admin Console welcome screen

5. Because the EJB beans use the DB2 UDB for iSeries database, you have to tell the WebSphere Application Server 6 test server where to find the Java objects that handle DB2 UDB for iSeries connectivity. You will need to add and create a Java Database Connectivity (JDBC) provider and data source to your WebSphere Application Server 6 test server configuration that will allow the EJB beans to interact with the iSeries database.

The First thing you will do is modify a WebSphere server environment variable name **OS400 TOOLBOX JDBC DRIVER PATH**. Expand the *Environment* category in the left navigator panel of the *Admin Console* and click on **WebSphere Variables**. (Figure 37)

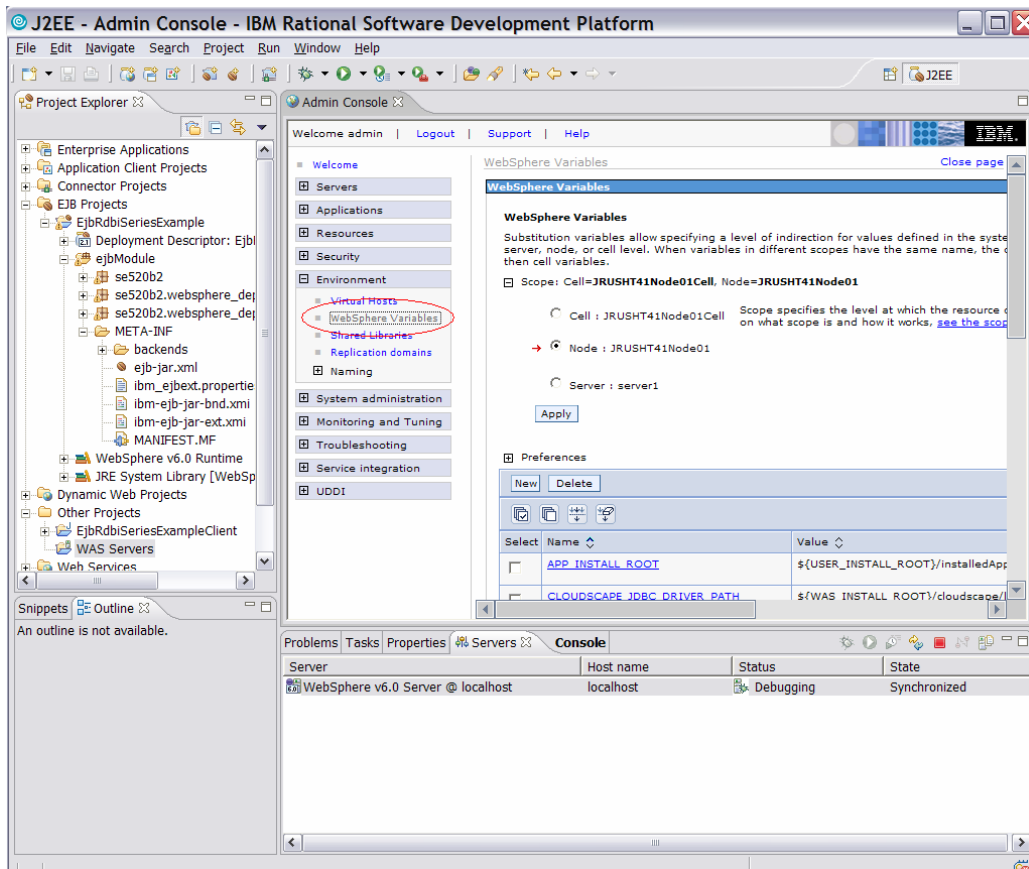


Figure 37: Admin Console; select WebSphere Variables

6. Pull the slider bar on the right side down until you see the **OS400 TOOLBOX JDBC DRIVER PATH** and click it. (Figure 38)

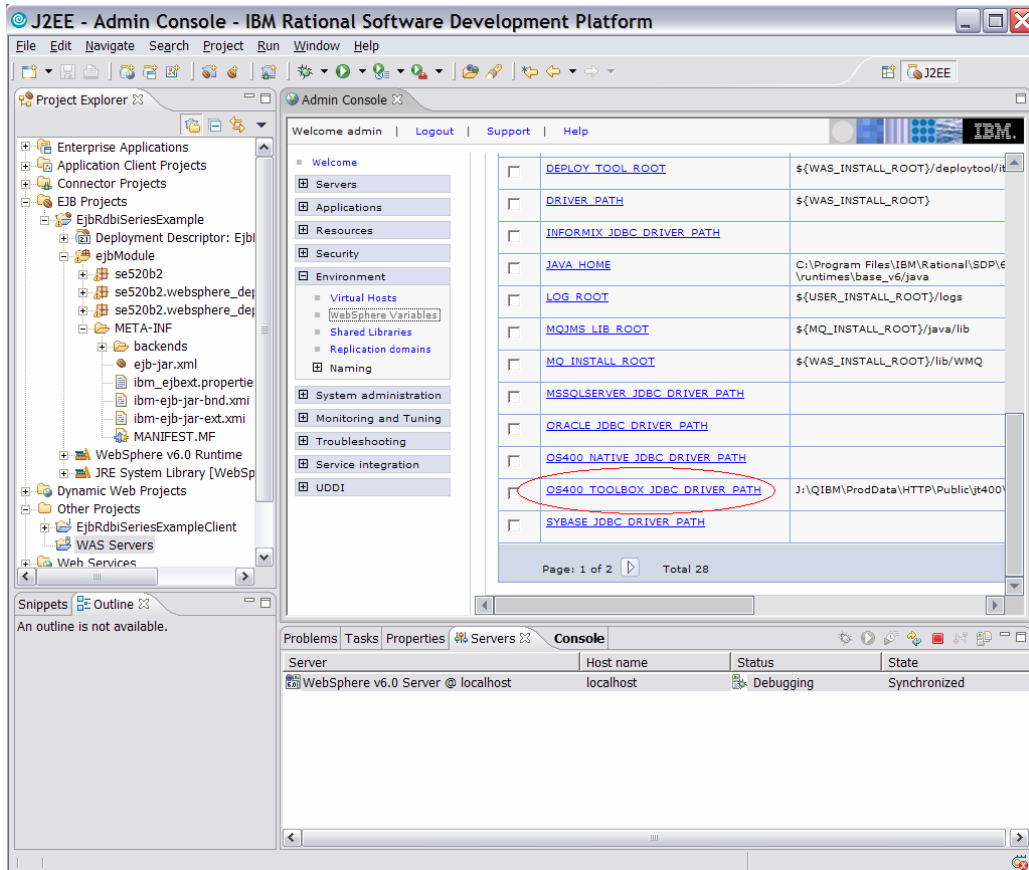


Figure 38: Select OS400 TOOLBOX JDBC DRIVER PATH

7. Type in the directory which contains the jt400.jar file on your workstation. This might be a local drive or, as in this example, a mapped network drive to the iSeries IFS file system. Click **OK**. (Figure 39)

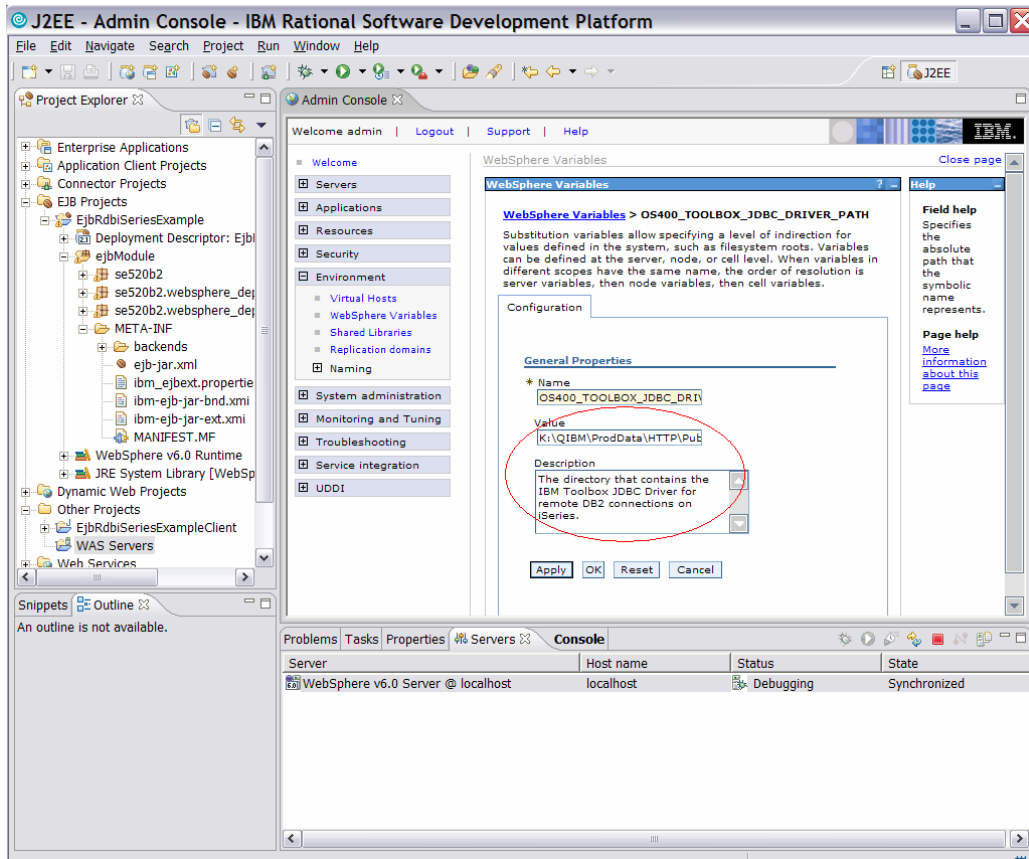


Figure 39: Select the correct directory

This will bring up a panel telling you the configuration has changed.

8. Click **Save configuration**.
9. Click **Save** on the next panel to save to the master configuration repository.

10. You will be brought back to the WebSphere variables panel and should see that the OS400 TOOLBOX JDBC DRIVER PATH has been updated. (Figure 40)

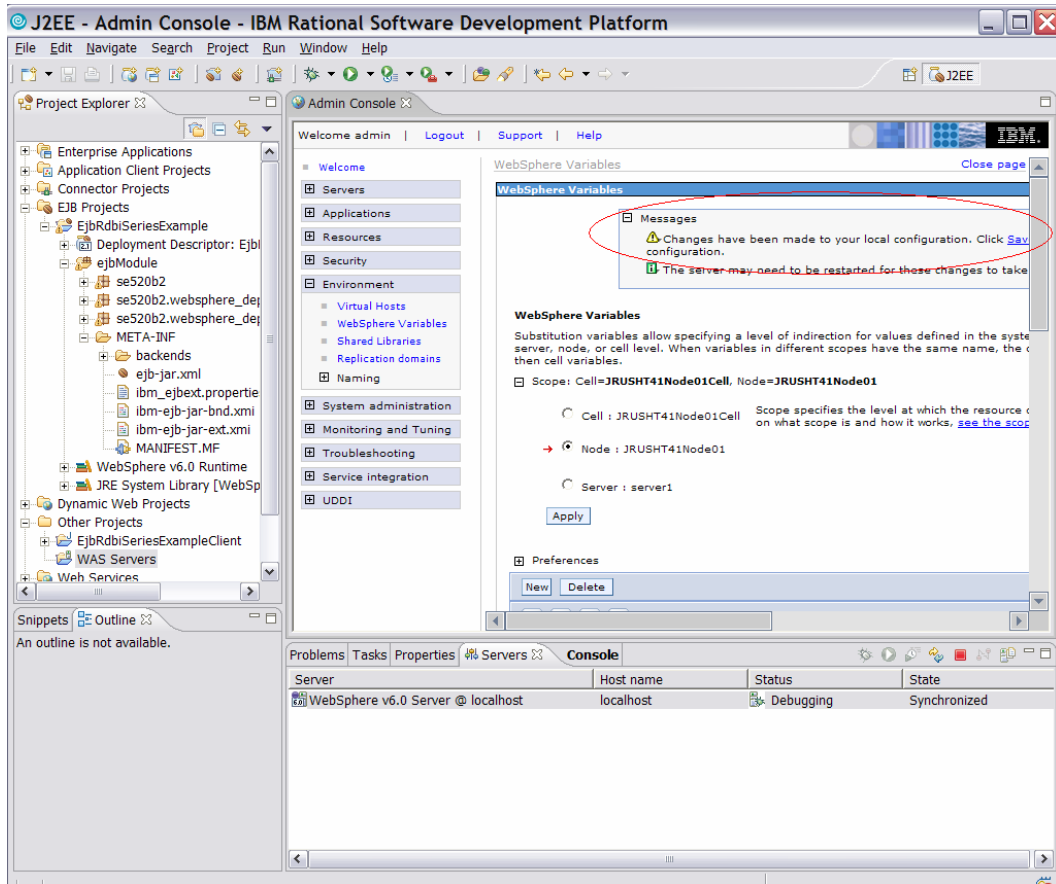


Figure 40: Driver modification message

Before we move on, let us maximize the Admin Console so we can see more of the administration panels without having to use the slider bars so often. In Rational Application Developer you can double click the tabs to maximize that tab.

11. Double click the **Admin Console** tab for an expanded view of the screen. (Figure 41)

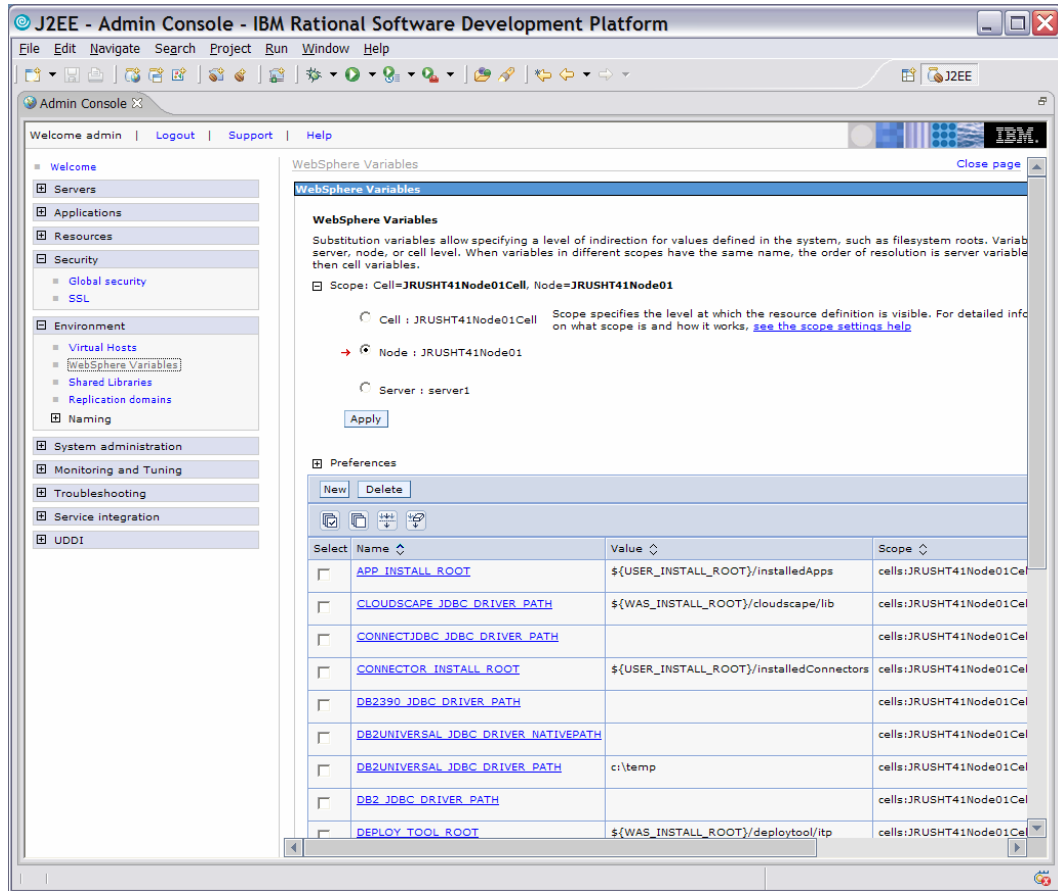


Figure 41: Expanded view of the Admin Console

12. Expand the *Security* category and then select **Global security**. (Figure 42)

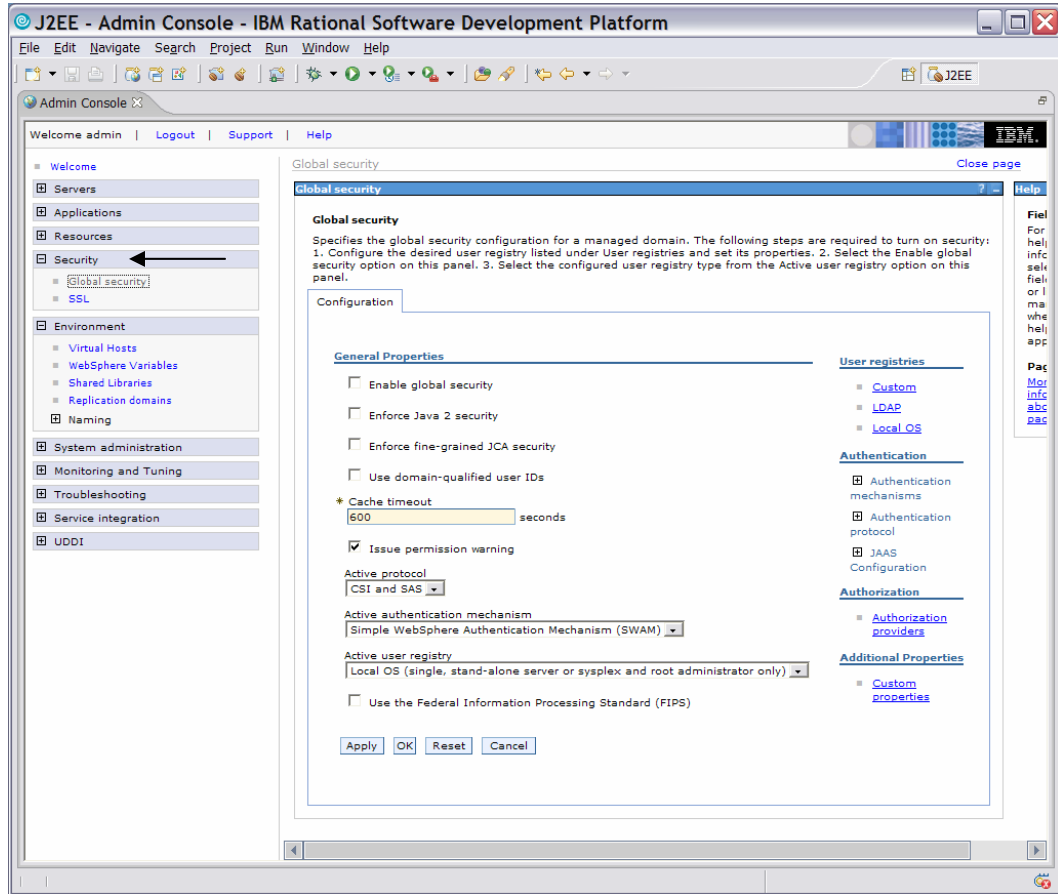


Figure 42: Global security

13. Expand *JAAS Configuration* and select **J2C Authentication Data** in the red circled area below. (Figure 43)

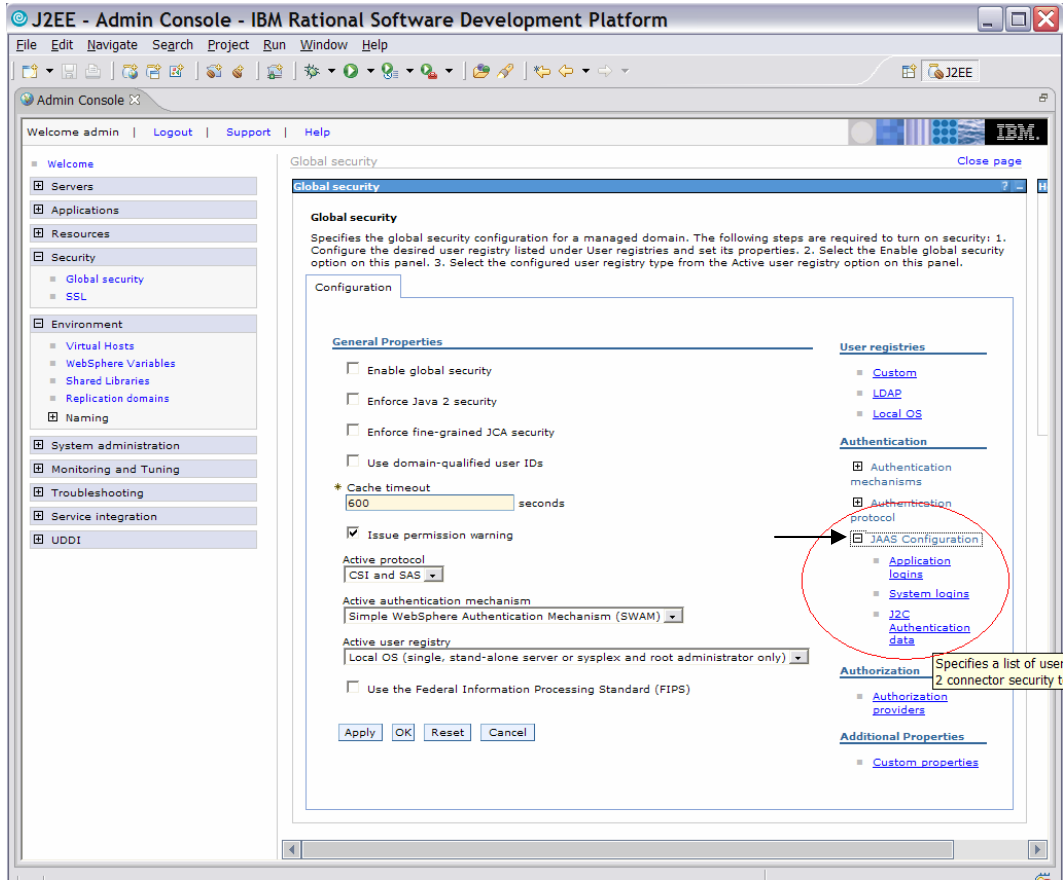


Figure 43: JAAS Configuration

14. In the Global security screen click **New**. (Figure 44)

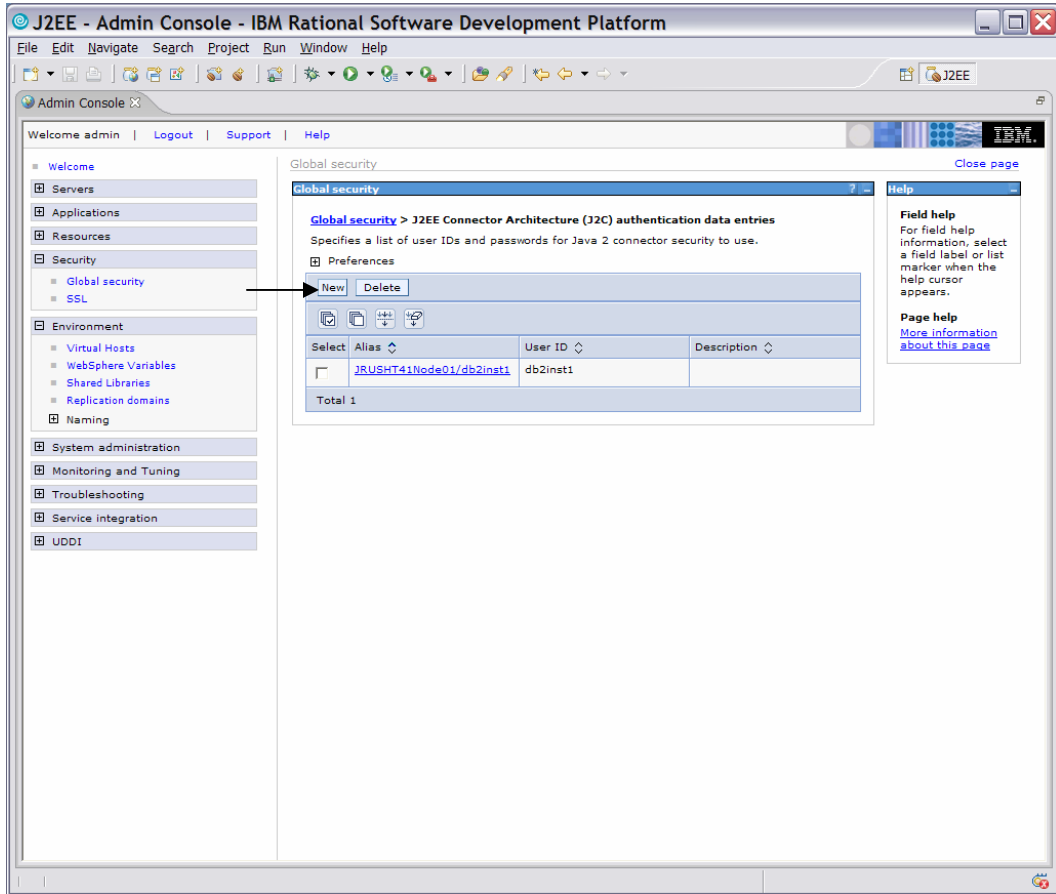


Figure 44: New Global security setting

This will open the **General properties** screen.

15. Enter your alias and user ID (jrush in this example) and password. **Note:** The user ID and password must be the same as the user ID and password of a valid user on the iSeries server where the database and tables being accessed reside (where the QIWS table resides in this example). (Figure 45)
16. Click **OK**.
17. Click **Save** to save changes to the local configuration.

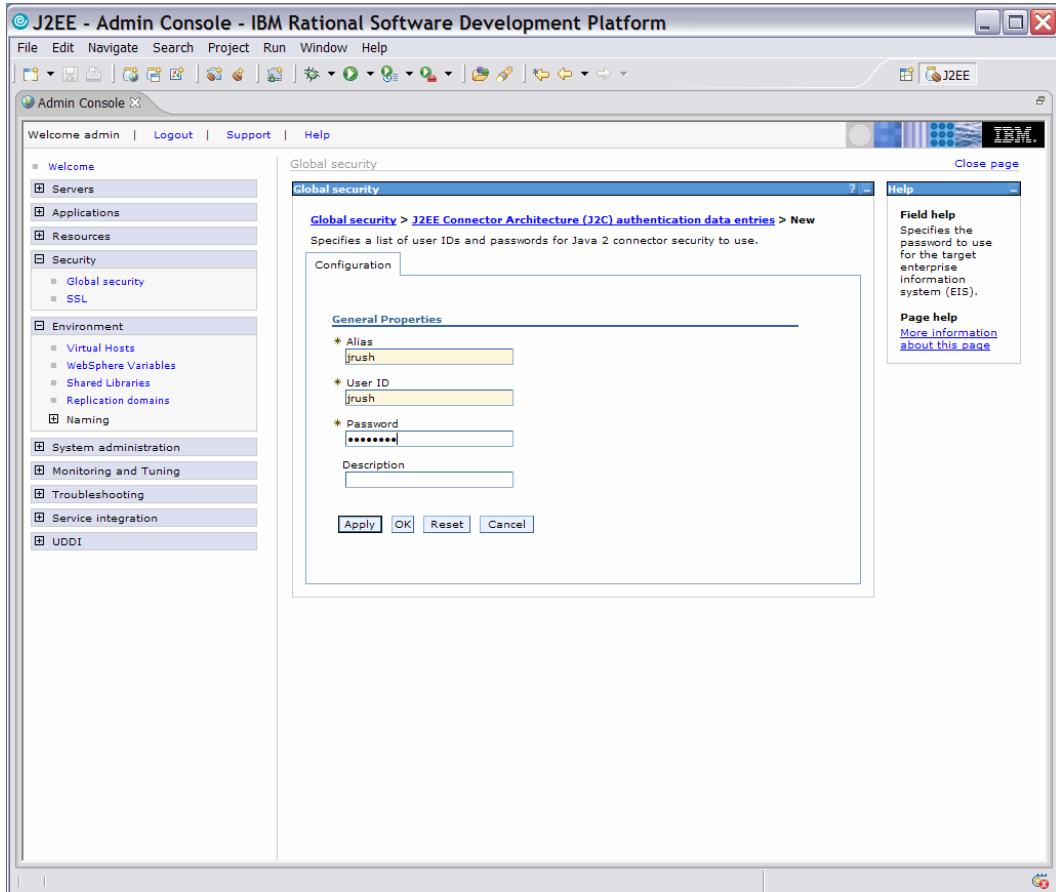


Figure 45: Authentication screen

18. After saving the configuration you will now see a panel like the one below with your **Alias** and **User ID**. (Figure 46)

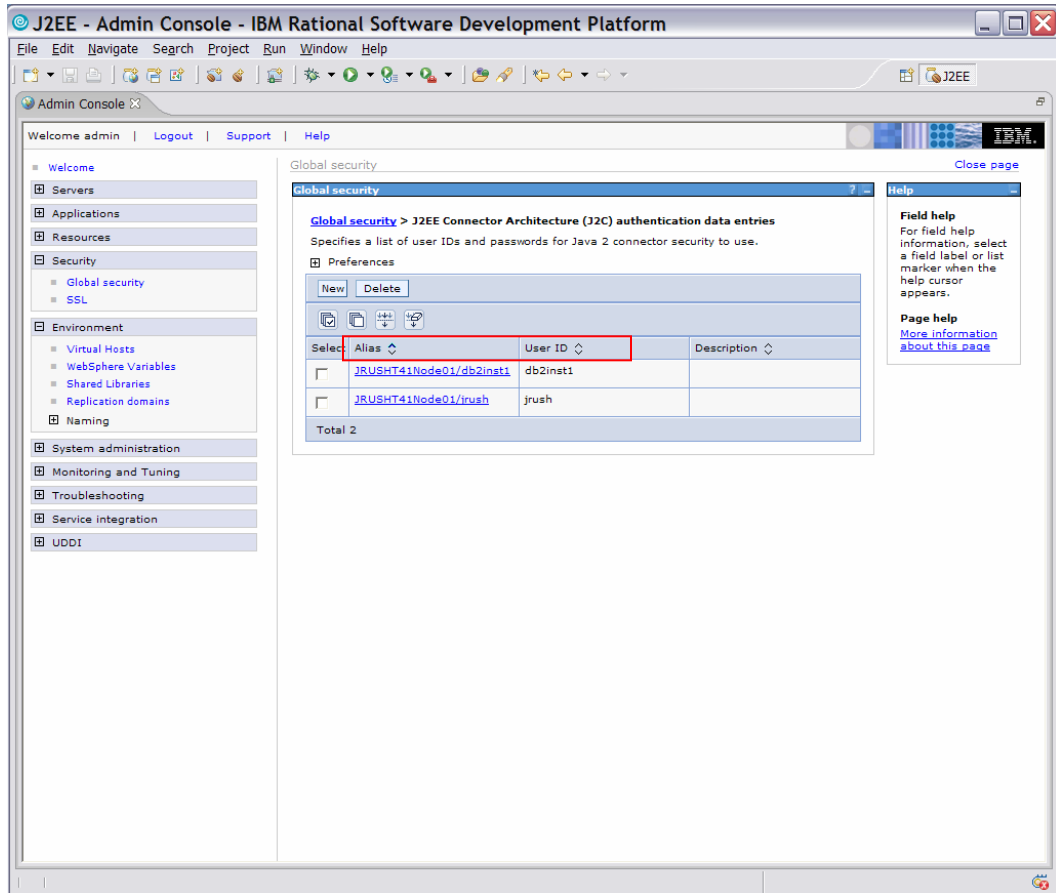


Figure 46: Aliases and user IDs

- Expand the *Resources* category and select **JDBC Providers** (Figure 47). You might see a list of providers already there. Cloudscape ships with the WebSphere Application Server.

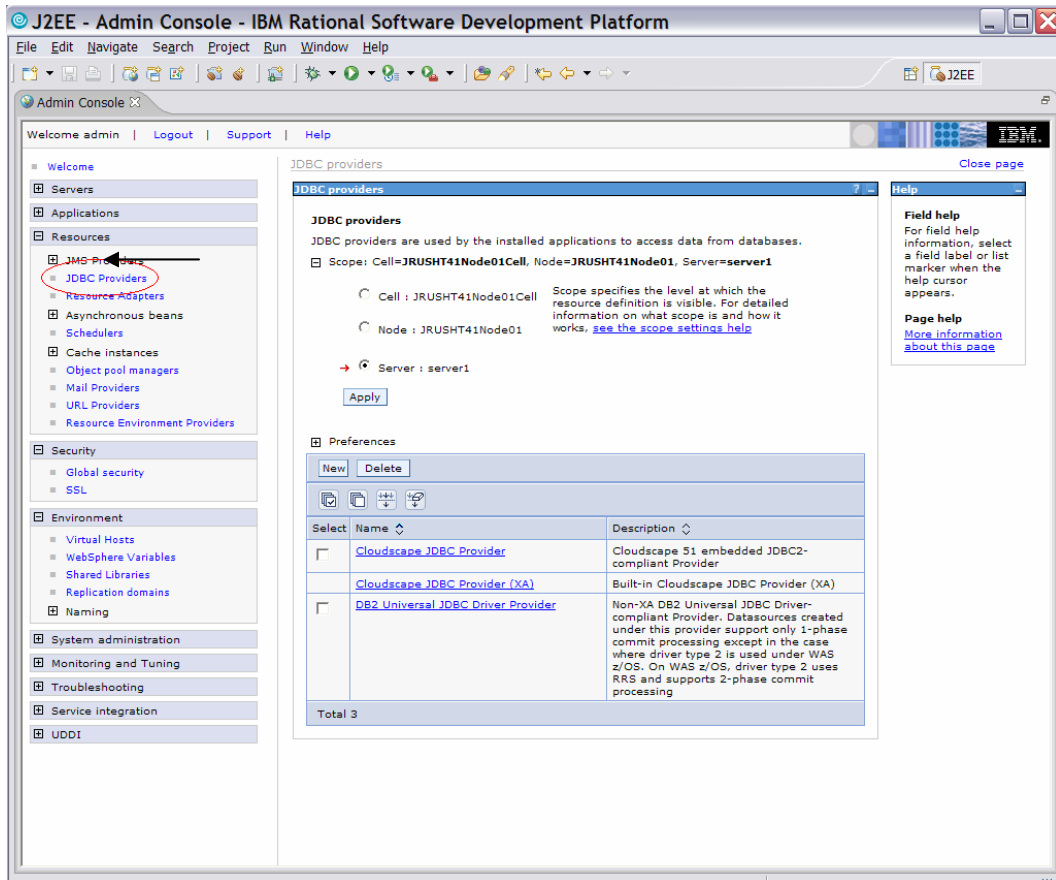


Figure 47: JDBC Provider resources

20. Select **New**. You are now going to define a database (JDBC) provider for the iSeries.
21. Under **Step 1**: select **DB2**.
22. Under **Step 2**: select **DB2UDB for iSeries (Toolbox)**.
23. Under **Step 3**: select **Connection pool data source**.
24. Click **Next**. (Figure 48)

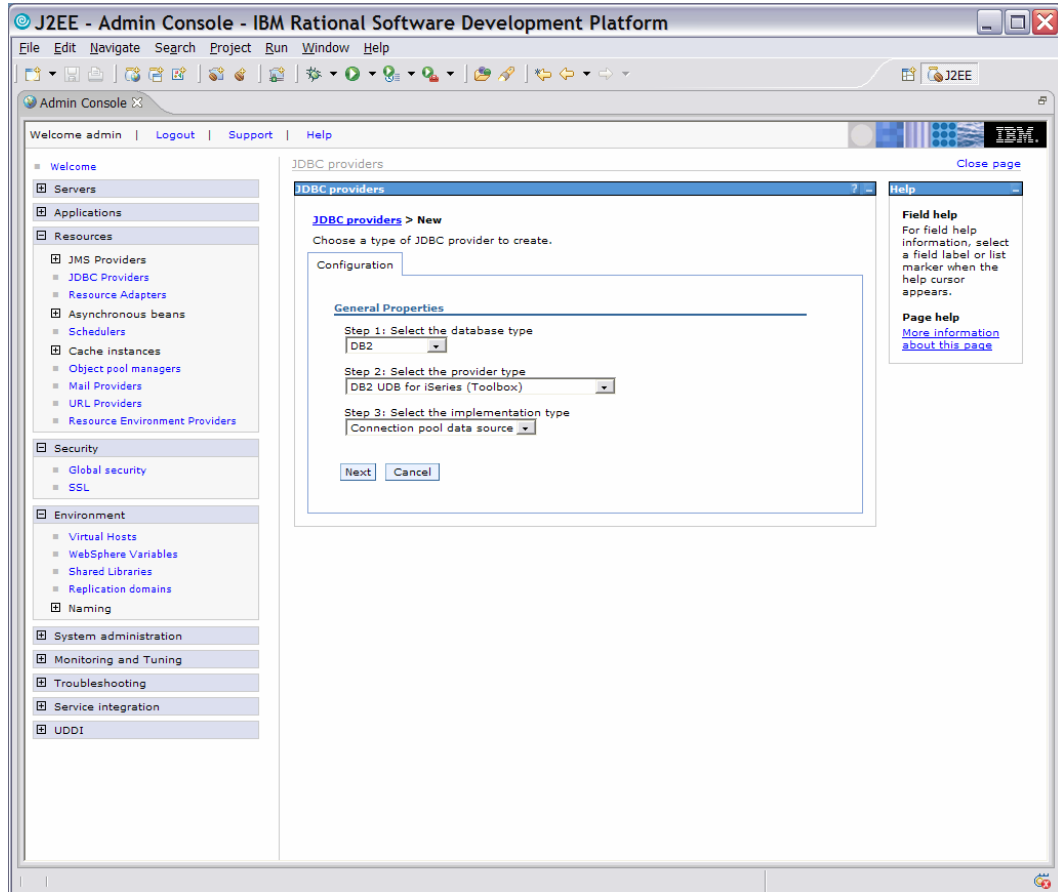


Figure 48: Set general properties

25. Check that the OS400_TOOLBOX_JDBC_DRIVER_PATH variable is set up properly. Everything on this panel should be OK.
26. At the bottom click **OK** (circled in red). (Figure 49)
27. **Save** your local configuration changes.

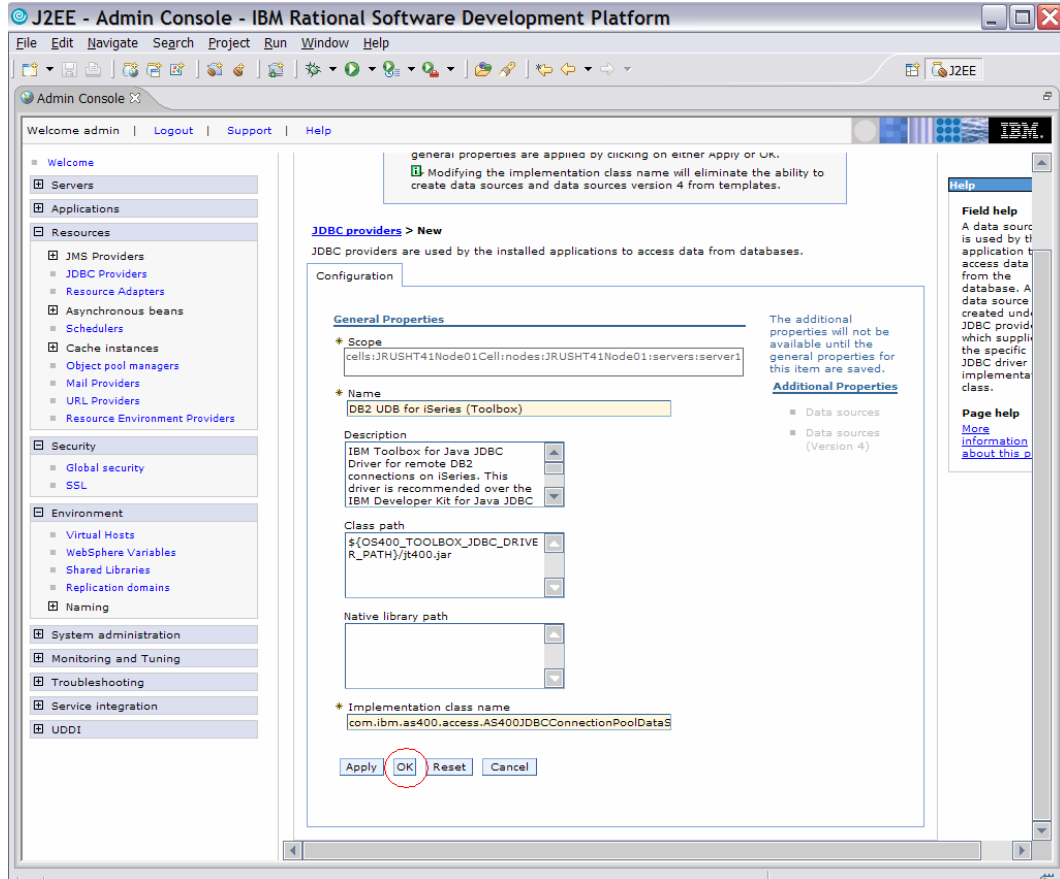


Figure 49: JDBC providers displayed

The new JDBC provider will now be displayed as one of the choices. (Figure 50)

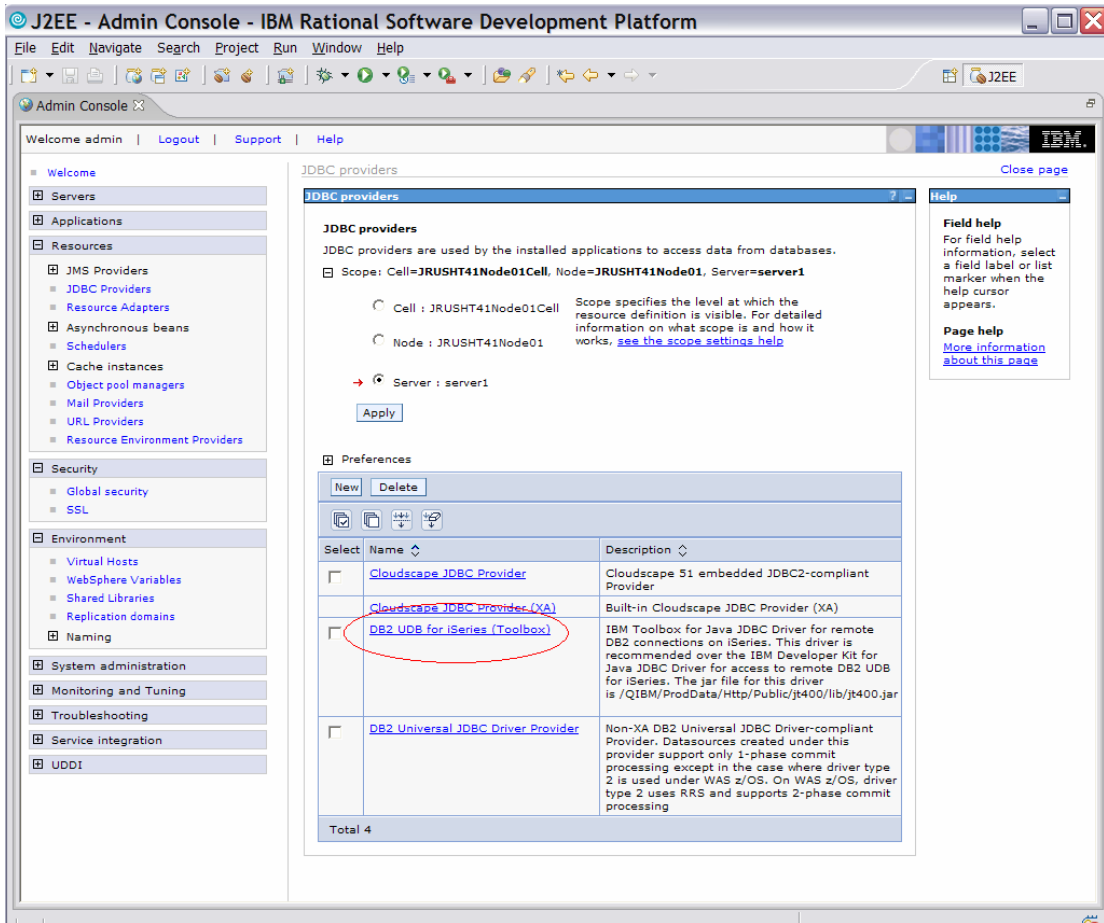


Figure 50: JDBC providers displayed

Add a data source or specific back-end iSeries server

The next process is to add a data source or specific back-end iSeries provider that will be connecting to the JDBC provider just created. In this example, you will create a Version 5 data source.

1. Click on the JDBC provider **DB2 UDB for iSeries (Toolbox)**. (Figure 51)

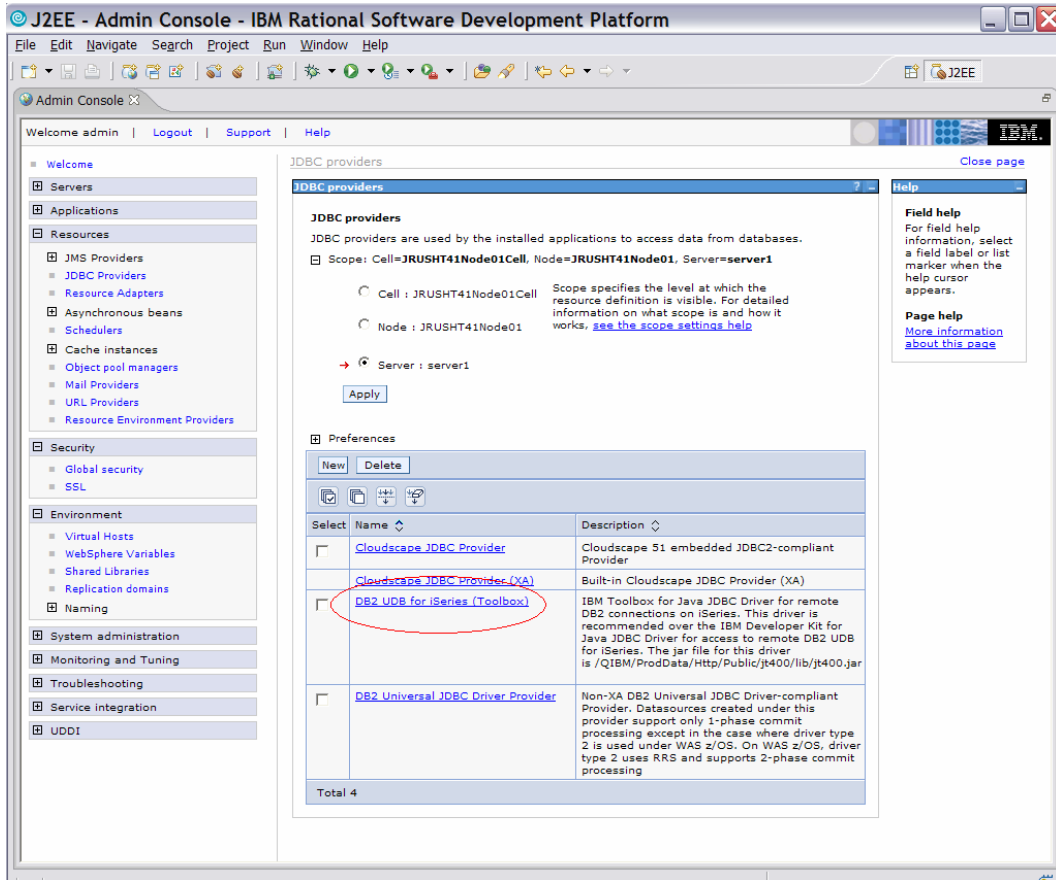


Figure 51: DB2 UDB for iSeries (Toolbox)

2. Under **Additional Properties**, click on **Data sources**. (Figure 52)

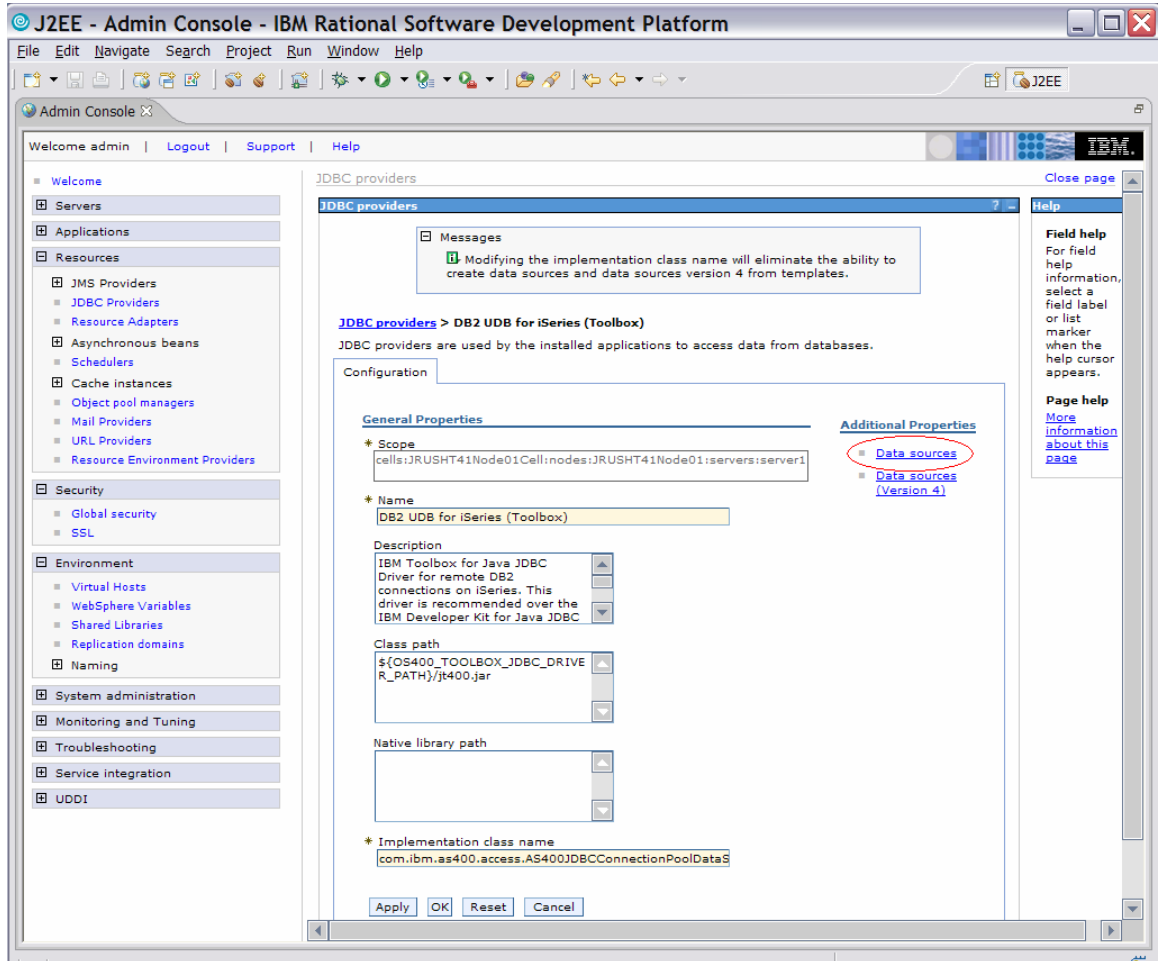


Figure 52: Data sources

3. Click **New**. (Figure 53)

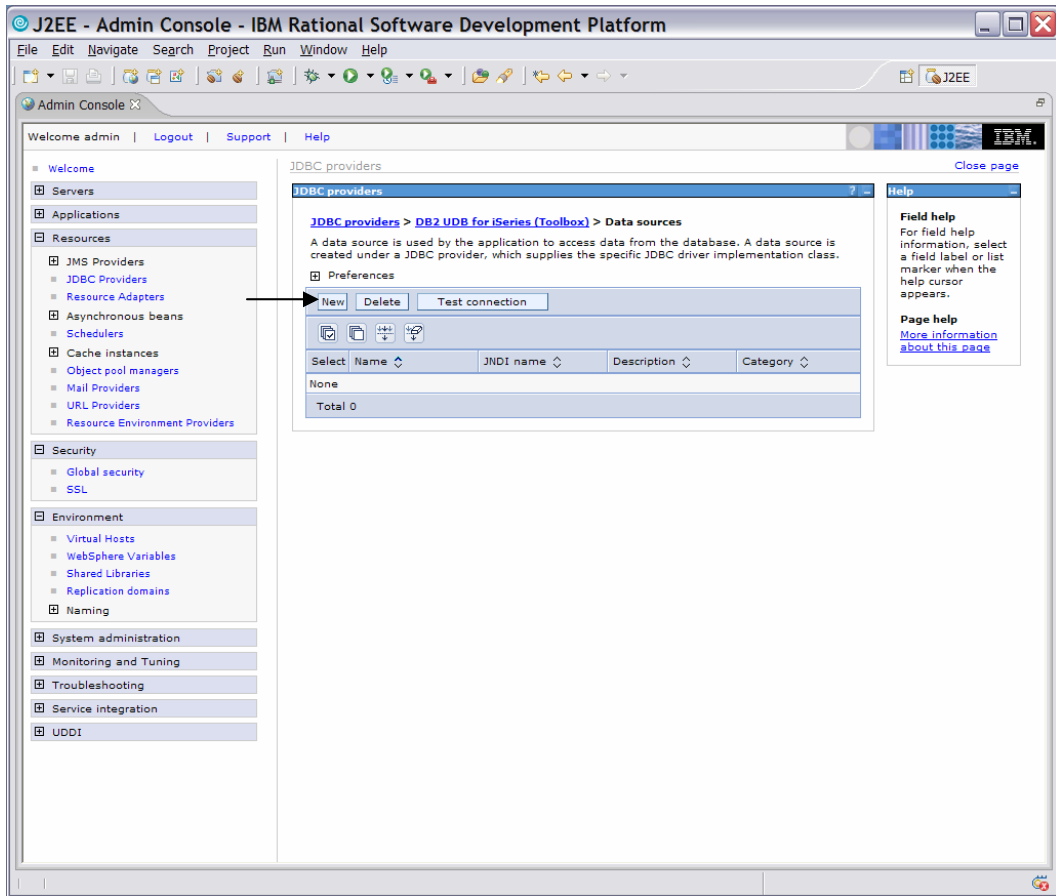


Figure 53: New data sources

- The following screen (Figure 54) shows the configuration fields for the iSeries data source. The JNDI name matches the updated EJB deployment descriptor *JNDI – CMP Connection Factory Binding* from earlier. The first image is the EJB Deployment descriptor and the second is the data source configuration panel.

Note that the circled JNDI names must match. The Component-managed authentication alias and Container-managed authentication alias are fields that have pull down menu selections based on the users that were created when the security tab of the integrated test server was selected. This user is used by the application server's JCA (Java Connector Architecture) security implementation when accessing the data source.

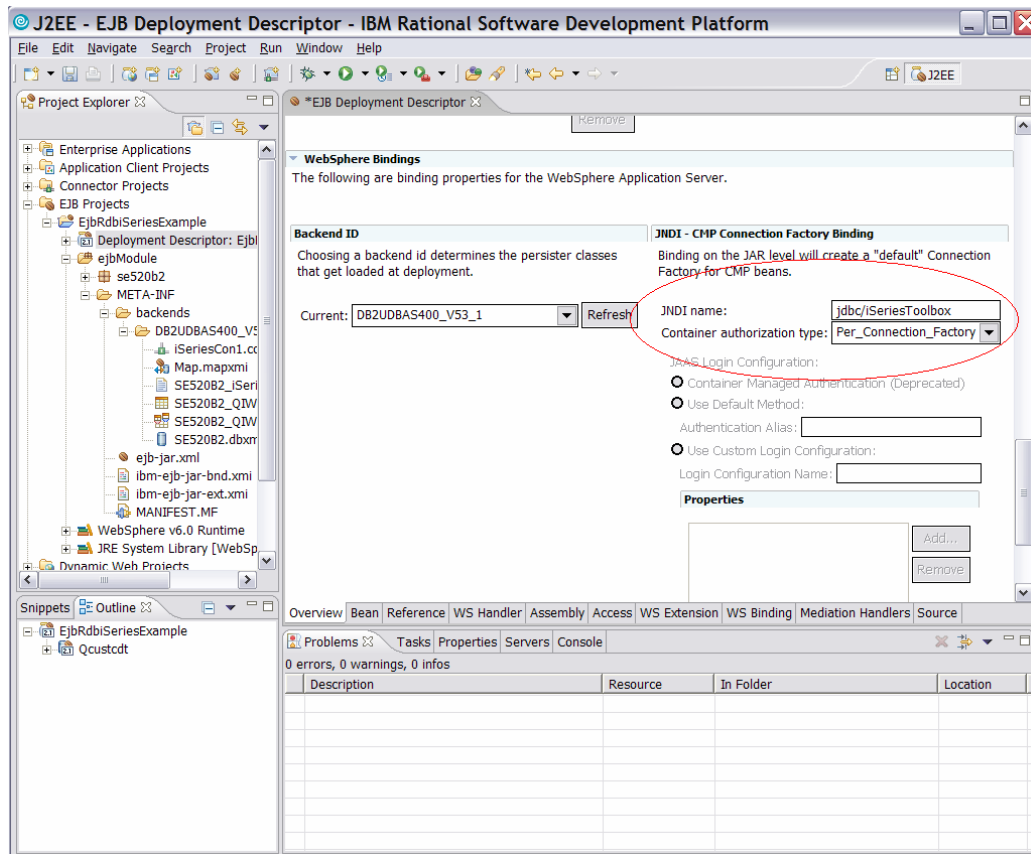


Figure 54: EJB Deployment Descriptor

5. When all fields are correct, click **Next**. (Figure 55)

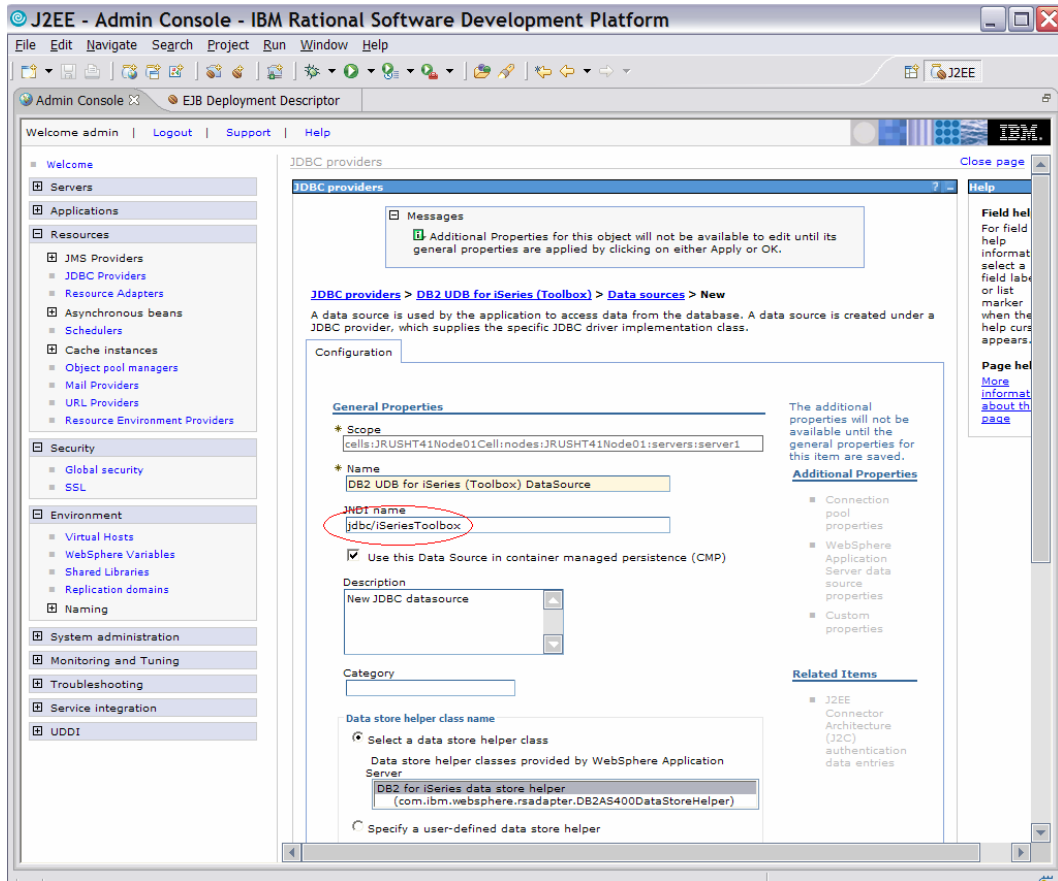


Figure 55: New data sources

6. The next step is to configure the security setting for **Component-** and **Container-** managed authentication. (Figure 56)

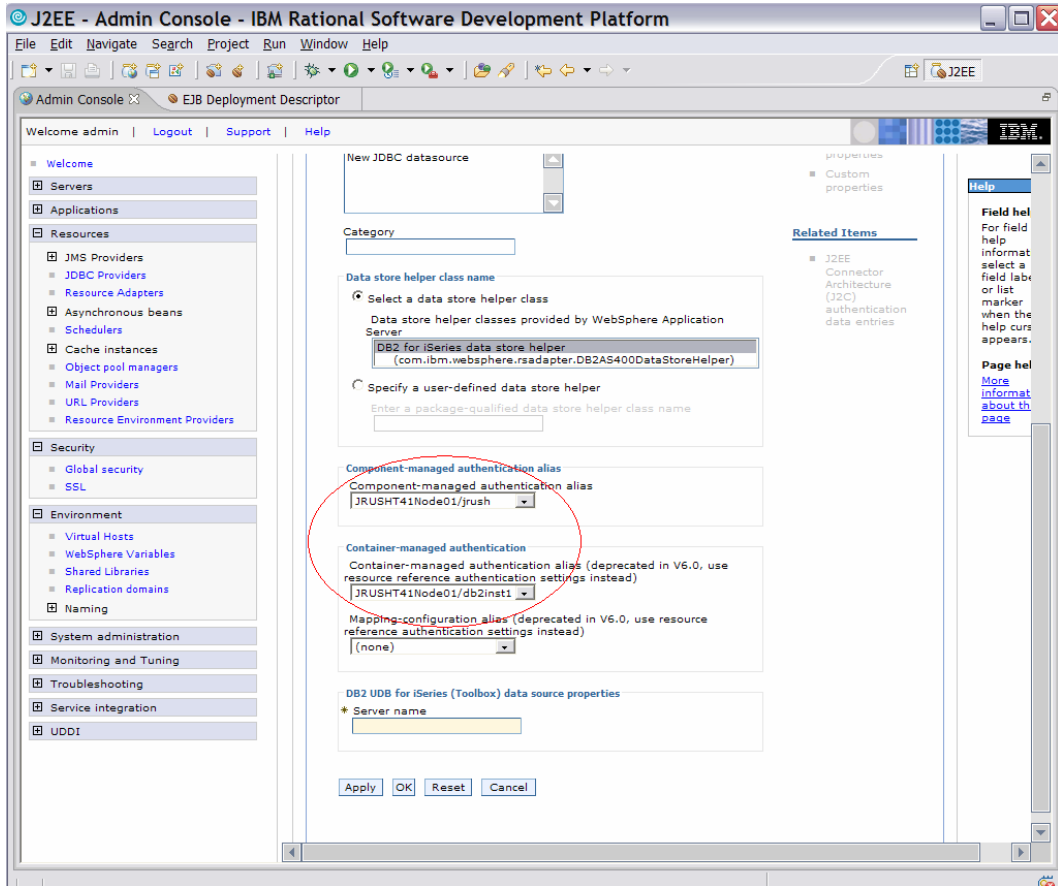


Figure 56: Managed authentication configuration

7. On the same panel, set the **Server name** field to your iSeries server. Click **OK**. Do not forget to save your configuration. (Figure 57)

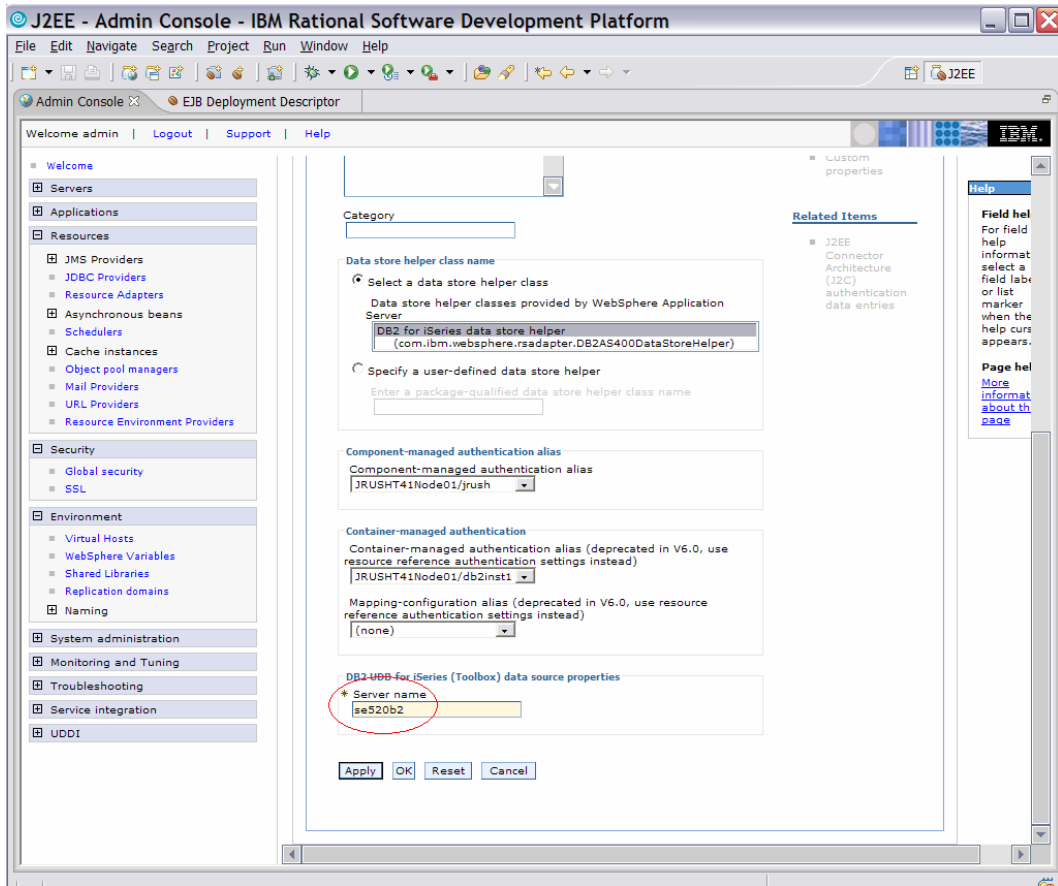


Figure 57: Server name configuration

8. Review the **Data sources** screen. (Figure 58)
9. Click **Logout** to log out of your session,

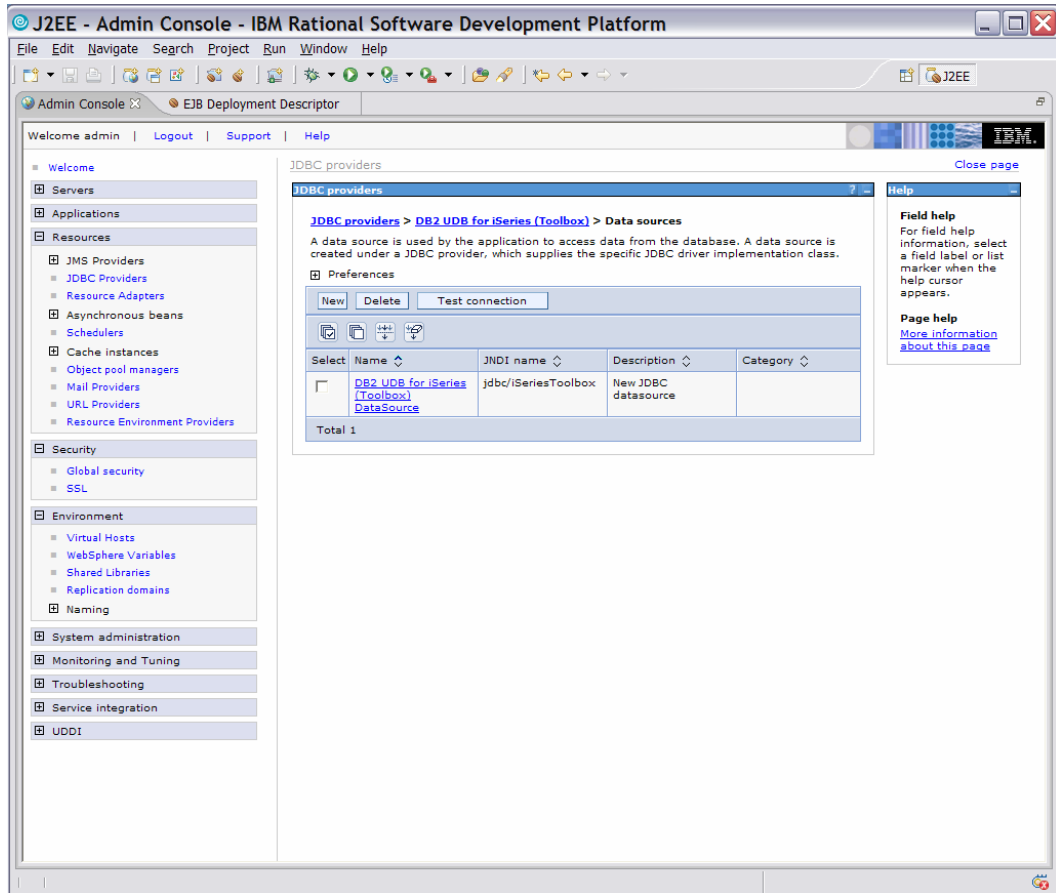


Figure 58: Final data sources screen

Test EJBs with test client

Rational Application Developer includes a test client application from which you can test the EJBs you create. In the final part of this lab, you will test the EJBs using this test client.

1. To start the test, right click on **EjbRdbiSeriesExample** and select **Run > Run on Server**. (Figure 59)

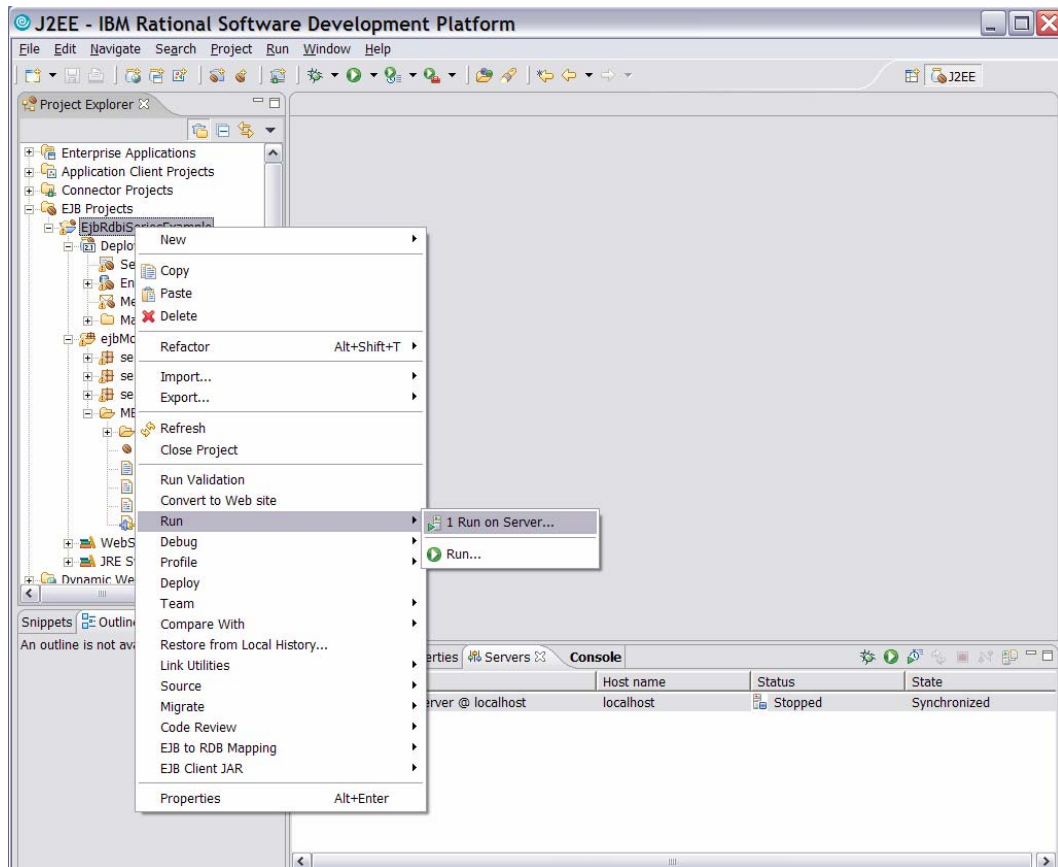


Figure 59: Run on server

2. Select **Choose an existing server** and then **EjbRdbiSeriesExampleServer**.
3. Click **Finish**.

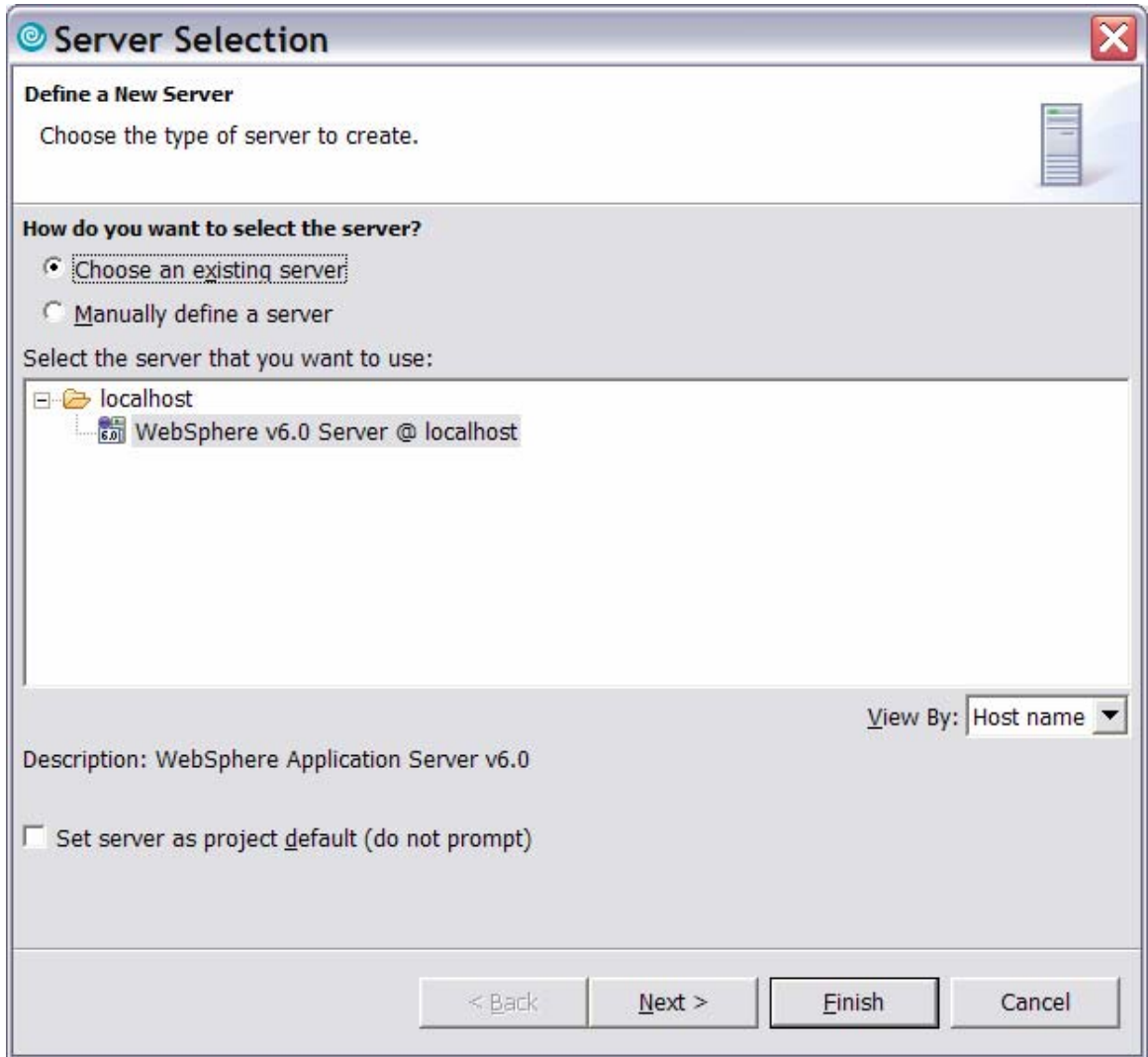


Figure 60: Choose an existing server, and finish

Note: The test server will now start. This might take some time.

4. The browser-based IBM Universal Test Client (UTC) will be displayed as shown below. Select **JNDI Explorer**.

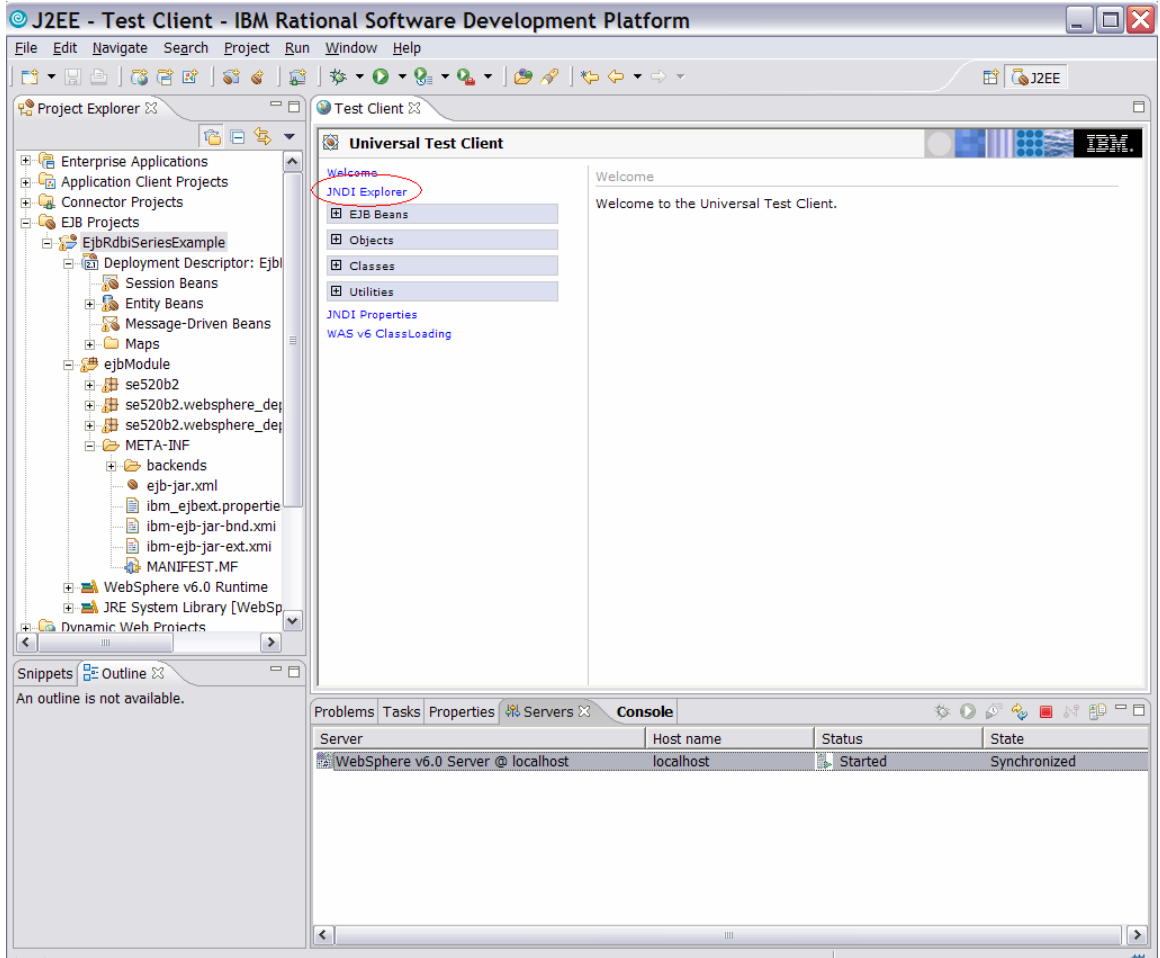


Figure 61: Select JNDI Explorer

- The **JNDI Explorer** panel allows you to see all the components that are currently bound and reside in the JNDI directory of the server. Select the **EJB Beans** option on the left side of the screen to display the EJB beans running on the Universal Test Client. (Figure 62)

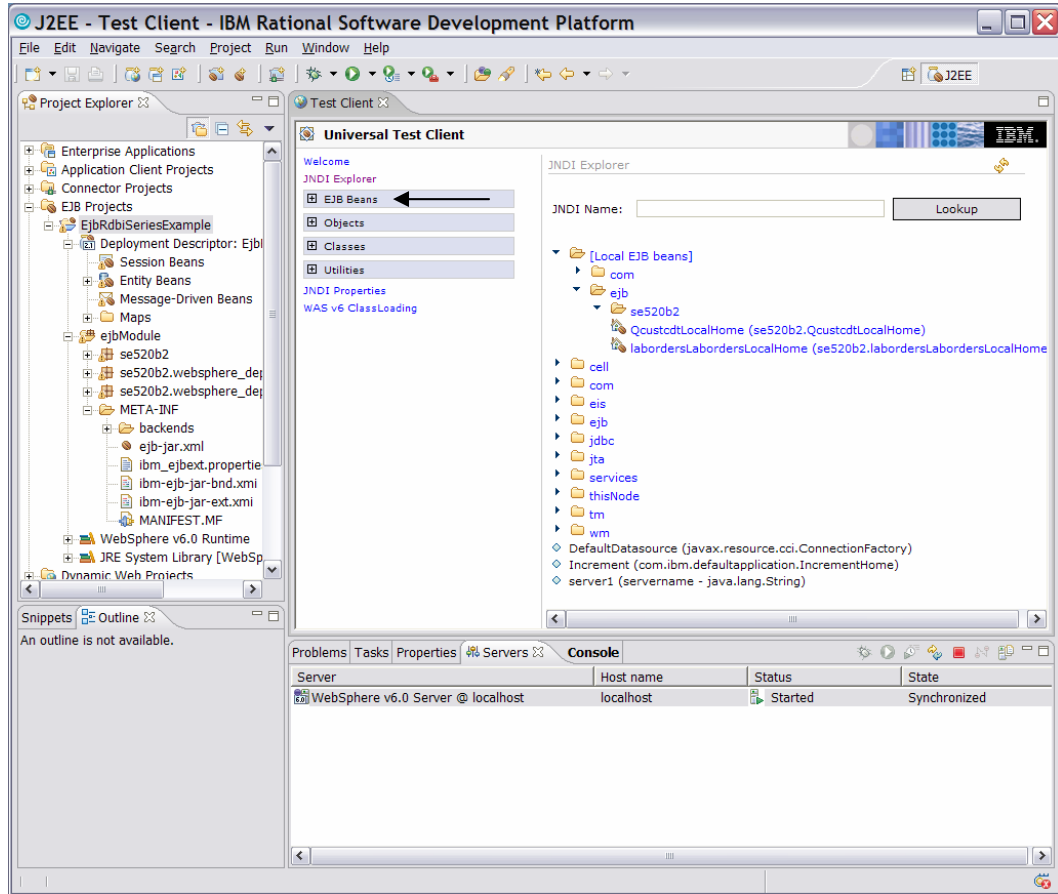


Figure 62: EJB beans running on the Universal Test Client

6. Click on the **QcustcdtLocalHome** bean.
7. On the next panel, expand **QcustcdtLocalHome** and then click **QcustcdtLocal findByPrimaryKey(QcustcdtKey)**. This allows you to interface with the remote server to instantiate an entity EJB with data from the iSeries database that has the customer number (CUSNUM), which you put in to the test client that will be used as the search criterion against the database. (Figure 63 in red.)

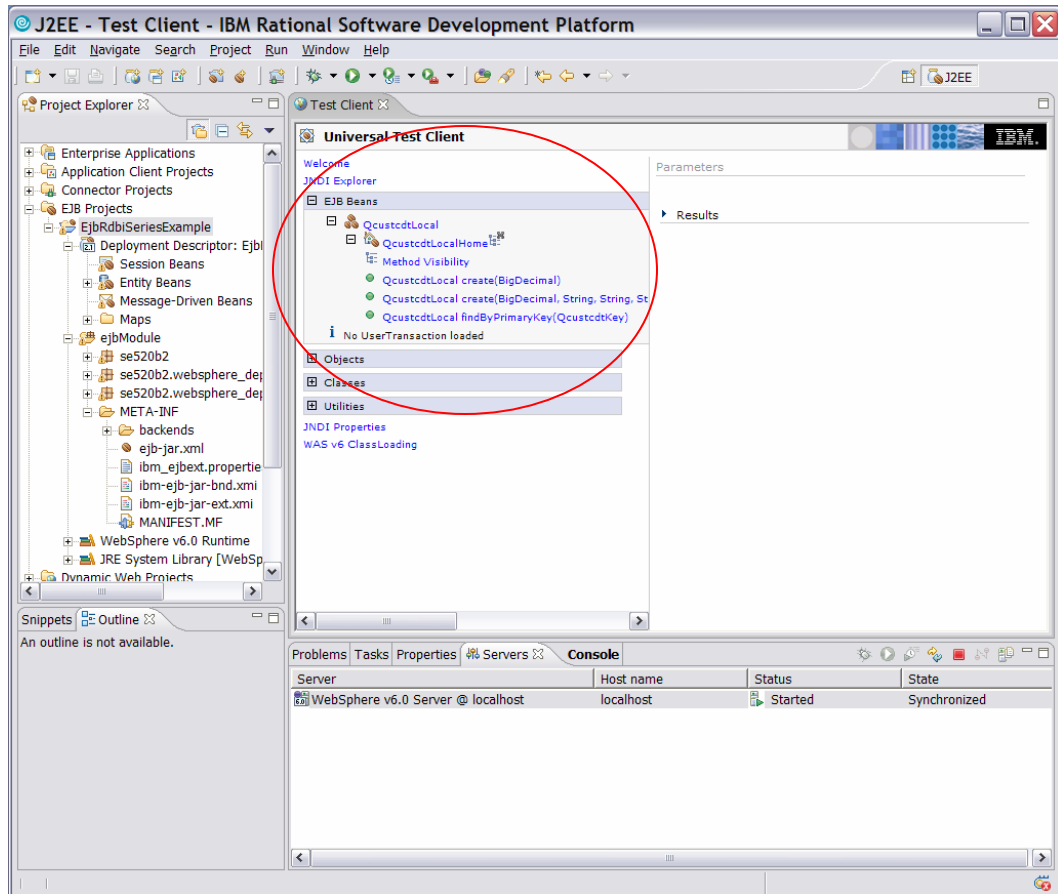


Figure 63: Universal Test Client console

8. Select **QcustcdtKey(BigDecimal)** from the second **Expand** pull-down menu. (Figure 64)

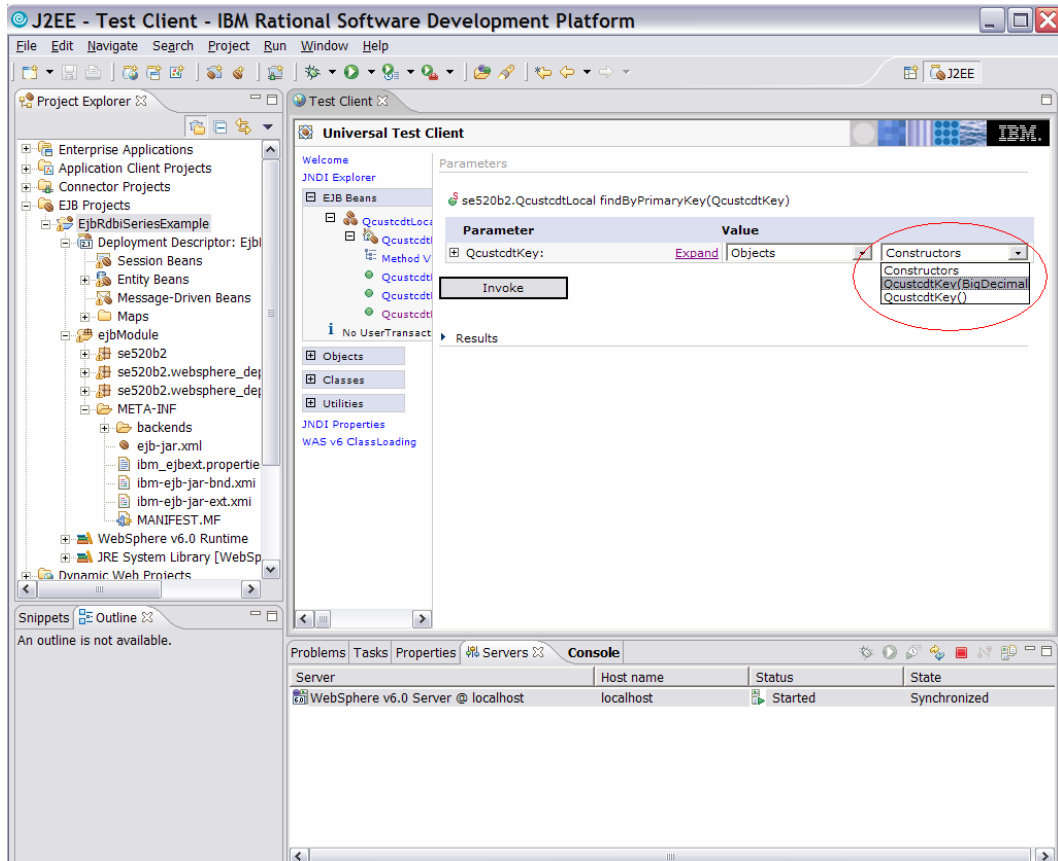


Figure 64: QcustcdtKey(BigDecimal) pull down menu

- Expand **QcustcdtKey** under *Parameter* and fill in **938472** under *Value* for the **BigDecimal** Parameter. Click **Invoke**. (Figure 65)

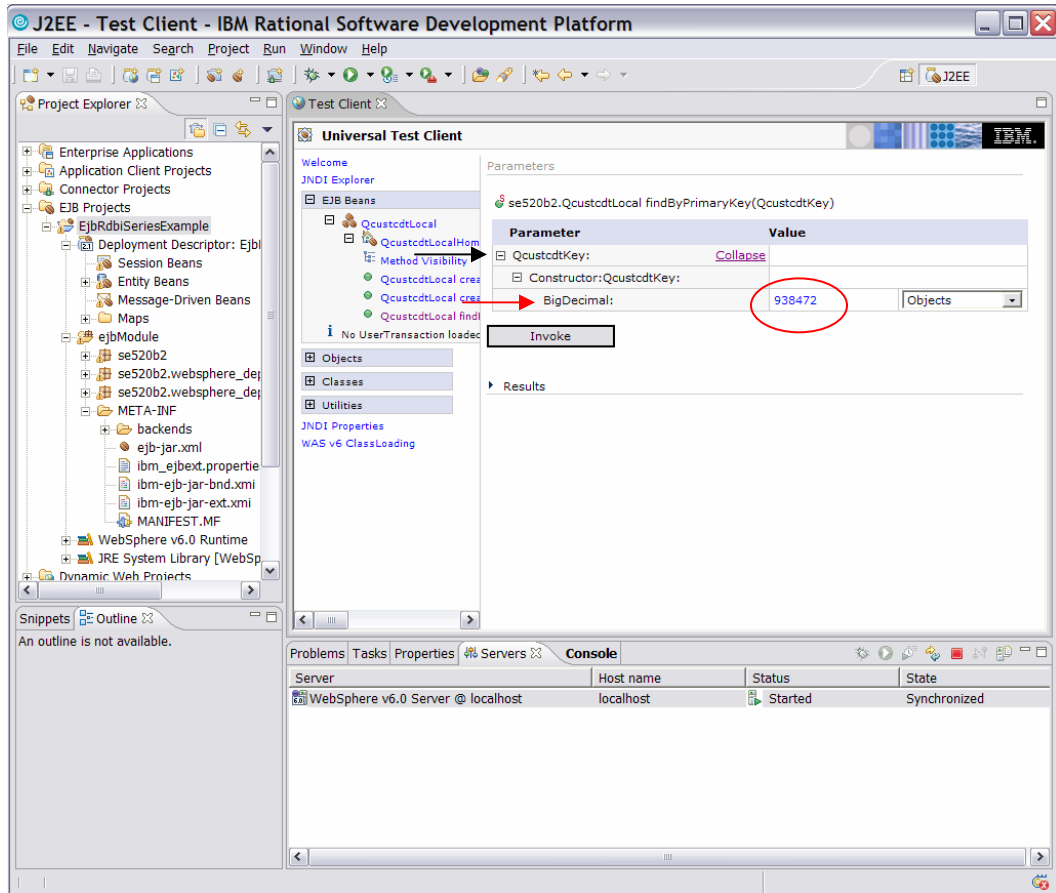


Figure 65: Invoke appropriate beans

10. Look at the *Results* section in the *Parameters* section.
11. Under *Results* click **Work with Object**. (Figure 66)

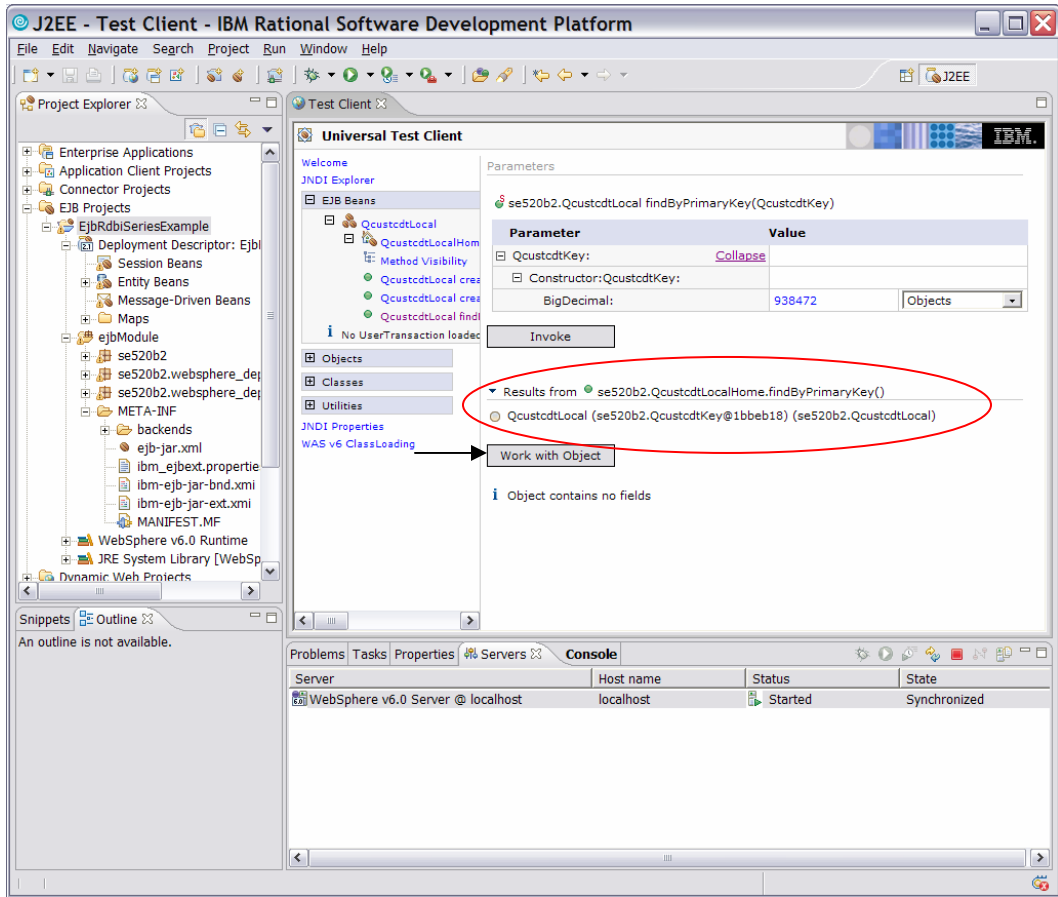


Figure 66: Work with Object

12. On the left side of the screen (Figure 67), you can see the methods available to interact with the entity EJB for this customer instance.
13. Select the **String** `getCity()` method.
14. Click **Invoke** in the right panel and you will see results of that method invocation.

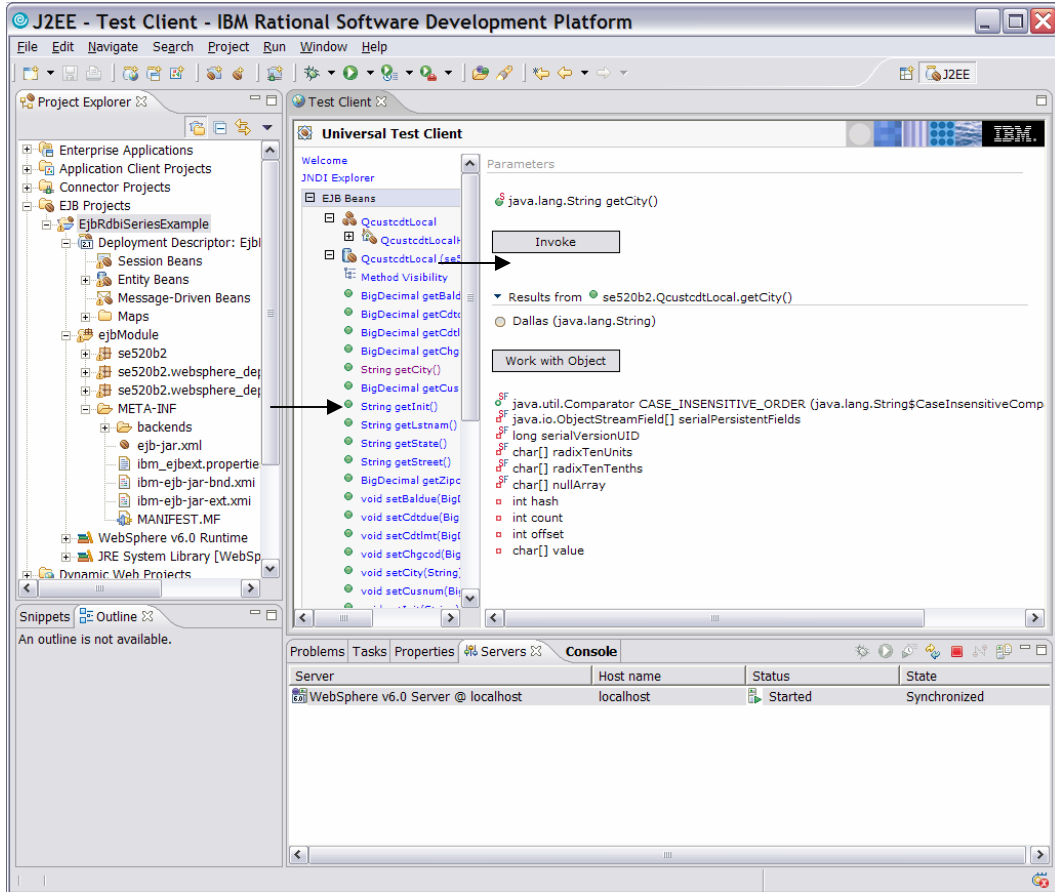


Figure 67: String `getCity()` method

Note: You have completed testing some of your EJBs. In a production environment, you would test all your EJBs in a similar fashion.

Summary

Rational Application Developer makes it easy to construct EJB artifacts that interact with mission critical data in the DB2 UDB for iSeries database. With its imbedded application server test environment and Universal Test Client, you can begin creating the building blocks of a robust enterprise application. You can use the EJBs created by this wizard as a starting point and further enhance them with more sophisticated business logic as it pertains to your environment.

Appendix A

Here is a more in-depth explanation for the different types of mapping for creating new EJB/RDB mapping:

Top-down

Assumes that you already have existing enterprise beans. In this approach, the enterprise bean design determines the database design. After you finish defining your enterprise beans, you can generate a schema and map.

A set of tables are generated to support the CMP entities inside the EJB project. In these tables, each column corresponds to a CMP field of the enterprise bean, and the generated mapping maps the field to the column. Relationships are mapped to foreign-key relationships.

By default, EJB inheritance hierarchies are mapped to a single table. In other words, the base and all derived enterprise beans are mapped to the same database table. Additional options exist that support generating joined tables for the leaf enterprise beans.

Note: If you do not manually map your beans, top-down maps are automatically generated when you generate deployed code using IBM DB2 UDB Version 7.2 as the default database vendor.

Bottom-up

This approach assumes that the database tables already exist, and that once the selected tables are imported, the enterprise beans and mappings between them are automatically generated. By default, relationships are generated where foreign-keys exist.

Meet-in-the-middle

In the meet-in-the-middle approach of mapping enterprise beans to database tables, it is assumed that you have existing enterprise beans and a database schema. Using the mapping editor, you can do a meet-in-the-middle mapping by matching by name, by type, or no matching.

In this approach, you map each field of the enterprise bean to the corresponding column of a table within the selected schema. Because roles have two ends, mapping one end (or one role) to a foreign-key, will automatically map the other end to the same foreign-key.

Additionally, if the source database or JAR file changes over time, the map and resulting tables or beans can keep up with these changes by "Re-executing" the mapping commands that are saved with the mapping document.

If there are any required maps missing, the task list will display an error for each missing map when the map file is saved, or validation is run on the project. This will prevent EJB deployed code generation from running until you fix the errors and complete the map.

Trademarks

© IBM Corporation 1994-2005. All rights reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both: IBM, the IBM logo, WebSphere, DB2, DB2 Universal Database, eServer, iSeries, xSeries, i5/OS, OS/400, AS/400, Toolbox for Java, and ibm.com.

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, other countries, or both.

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

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

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

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

Information is provided "AS IS" without warranty of any kind.

Other company, product, and service names listed throughout this presentation are the trademarks of their respected providers.

About the Author

Jon Rush

Senior Software Engineer
IBM eServer Solutions

Jon Rush is a Senior Software Engineer in the IBM eServer Solutions Enablement organization. He focuses on helping ISVs enable their WebSphere applications on Linux™ on POWER.