

## **Oracle EBS Adapter Sample – Calling Oracle Native API**

This edition applies to version 6, release 1, modification 0 of IBM® WebSphere® Adapter for Oracle E-Business Suite on WebSphere Application Server (product 5724-T73) and to all subsequent releases and modifications until otherwise indicated in new editions. To send us your comments about this document, email <mailto://doc-comments@us.ibm.com>. We look forward to hearing from you. When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you. © **Copyright International Business Machines Corporation 2007. All rights reserved.** US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

## Introduction

This sample shows how the Adapter for JDBC can call an Oracle E-Business Suite API directly. The API that will be called is APPS.FND\_PROGRAM.EXECUTABLE(). It is used to create a concurrent program executable in Oracle EBS.

This sample uses the Adapter for JDBC to call an Oracle API directly, which creates a concurrent program executable in the Oracle database. The content for this sample includes examples of the API call.

The sample assumes you have experience with the Oracle EBS software. Only general guidelines are given for steps that must be performed in that software.

*Note: The JDBC driver limits parameters to simple, non-record data types when the Adapter for JDBC calls an API directly. Most Oracle APIs include record parameters. API calls that use record type parameters can be called through the Adapter for JDBC only if they are wrapped so that the call to the wrapper stored procedure does not use the record type parameter. The API call for this sample uses simple data type parameters.*

## Database and application user account requirements

To use the samples, you must use a database account that gives you rights to the artifacts needed to run the sample content, and use an Oracle E-Business Suite account that allows you to perform responsibilities of the System Administrator and Receivables Manager.

For the purposes of these samples, the user account running all scripts is assumed to be the APPS user for the Oracle database. This user has the following rights:

- To change and create content in the APPS schema
- To add and remove data from tables
- To run the required executables in the APPS schema

Check with your Oracle database administrator to determine the account that will be used to run the sample content. If you want to choose a different user account, work with your database administrator to ensure that the user has rights to all of the database artifacts needed to run the sample content.

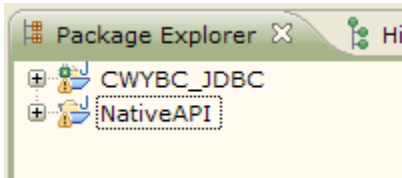
Oracle E-Business Suite requires you to have an account with rights to the following responsibilities:

- System Administrator
- Receivables Manager

Note: If the Oracle E-Business Suite account that you are using does not have access to these responsibilities, they can be added. To change the responsibilities assigned to the account you will use, log onto an account that has System Administrator responsibility rights and go to the Security->User->Define menu option. For specific information about changing user responsibilities, refer to your documentation for Oracle Applications.

## Running the prepared sample

Import the project interchange file containing the OracleEBS API sample, your workspace should look as follows:



Now you must add the jdbc driver library to your CWYES\_JDBC connector project. Add the file to the project using build path configuration. From the popup menu select configure build path, then select Libraries tab and press Add External JARs... button. Navigate to the location of the JDBC driver and press Open, then Ok. Expand NativeAPI project and select EBSAPIImpl class in the nativeapi.executable package.



At the beginning of the file, edit the annotations pointing your Oracle EBS system. You must provide the following information:

- Database URL – the database URL string
- User Name – valid user name to access OracleEBS system
- Password – valid password for the above user. The password is stored in plain text and therefore it should only be used temporarily for the sample. For the production deployment, the security alias mechanism should be used.

```

/**
 * @j2c.managedConnectionFactory class="com.ibm.j2ca.jdbc.JDBCManagedConnectionFactory"
 * @j2c.managedConnectionFactory-property
 *     name="databaseURL" value="jdbc:oracle:thin:@my_host:1521:my_database"
 * @j2c.managedConnectionFactory-property name="databaseVendor" value="ORACLE"
 * @j2c.managedConnectionFactory-property name="jdbcDriverClass" value="oracle.jdbc.driver.OracleDriver"
 * @j2c.managedConnectionFactory-property name="password" value="password"
 * @j2c.managedConnectionFactory-property name="returnDummyBOForSP" value="false"
 * @j2c.managedConnectionFactory-property name="userName" value="user"
 * @generated
 */

```

Once you edit the annotations and save the file, the code is regenerated to reflect the new values and the program is ready to run.

Select the client.ApplicationClient file and from the popup menu select Run As, then Java™ Application.

### Verify the results using the Oracle EBS web client

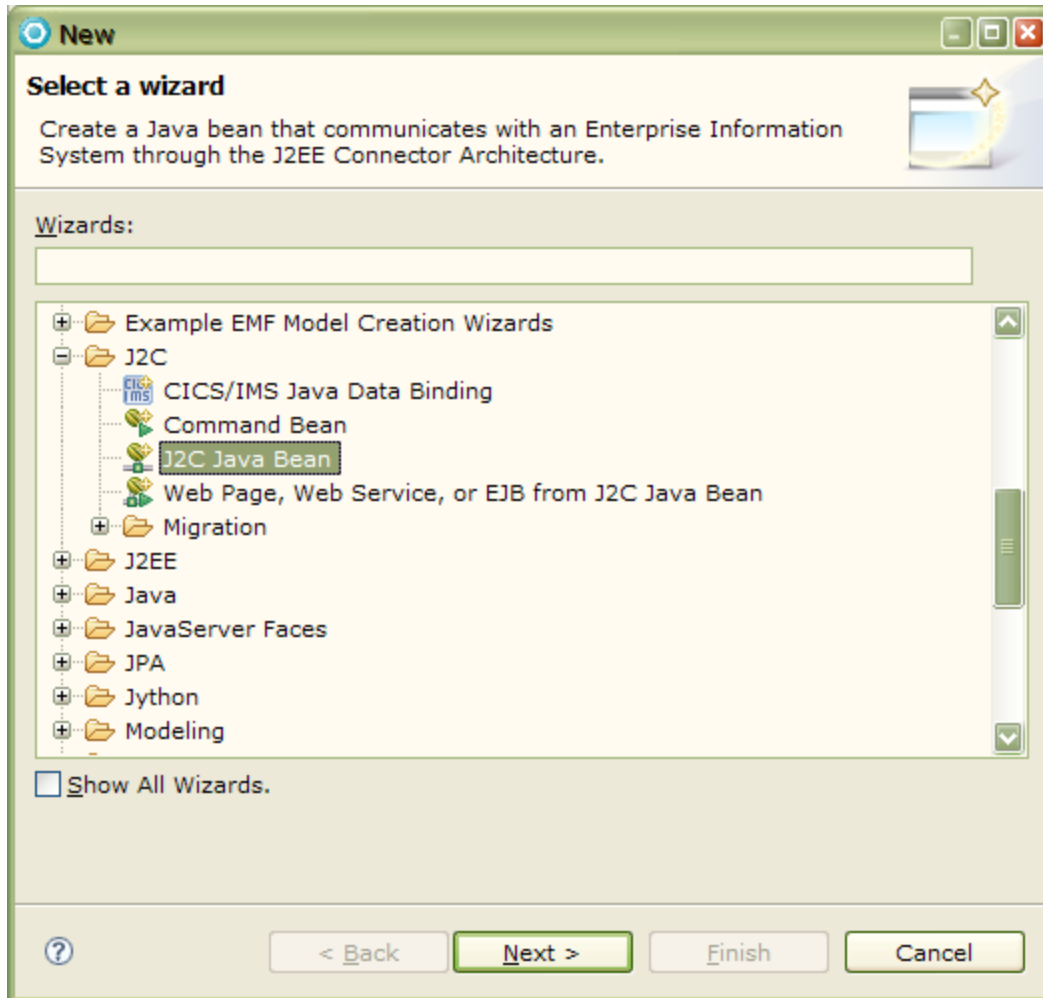
- Log into Oracle EBS.
- Select System Administrator responsibility.
- Select Concurrent -> Program option. This will bring up a Concurrent Program screen.
- Query the window for the executable.
- In the Short Name field, type 'IBMSAMPLENAME'. Click Go.
- You should see the executable details listed.

### Building the complete sample.

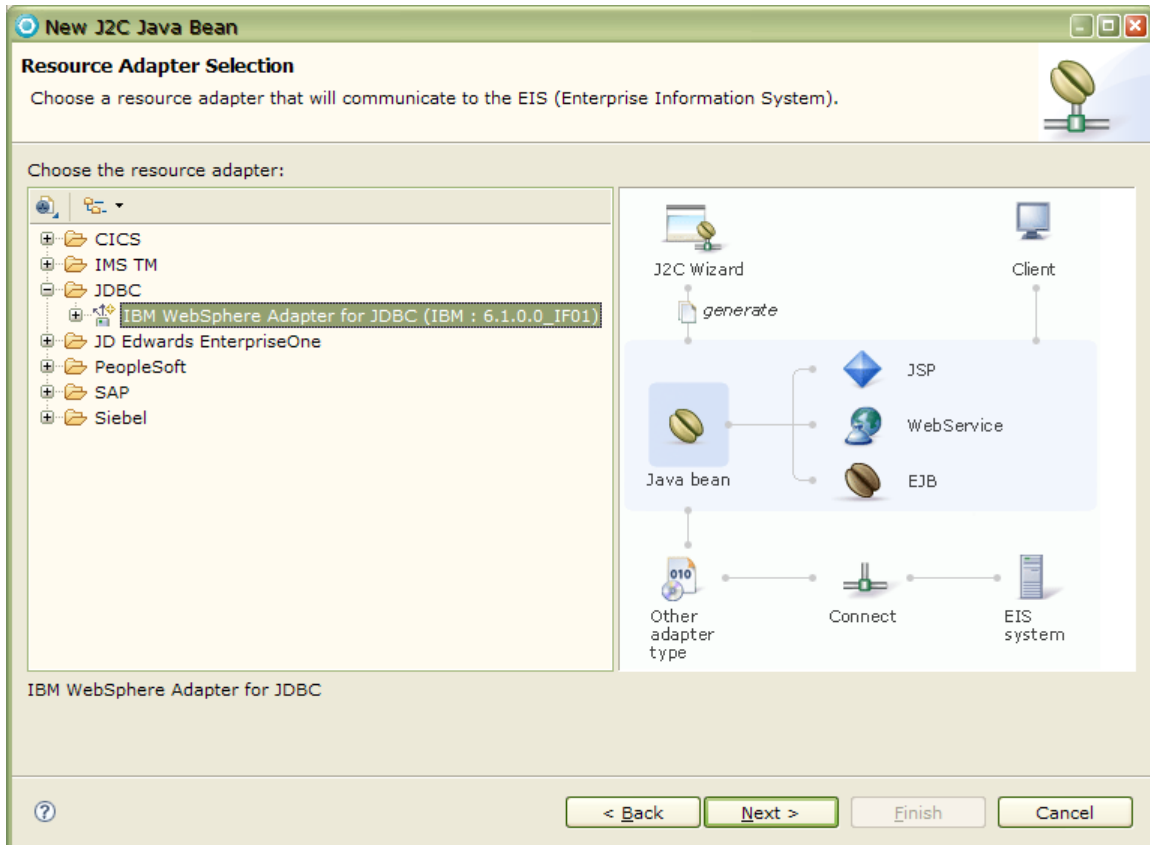
In this part of the sample, we' will walk through the complete steps of building and running the OracleEBS Native API sample.

#### Create J2C Java Bean accessing OracleEBS Native API system

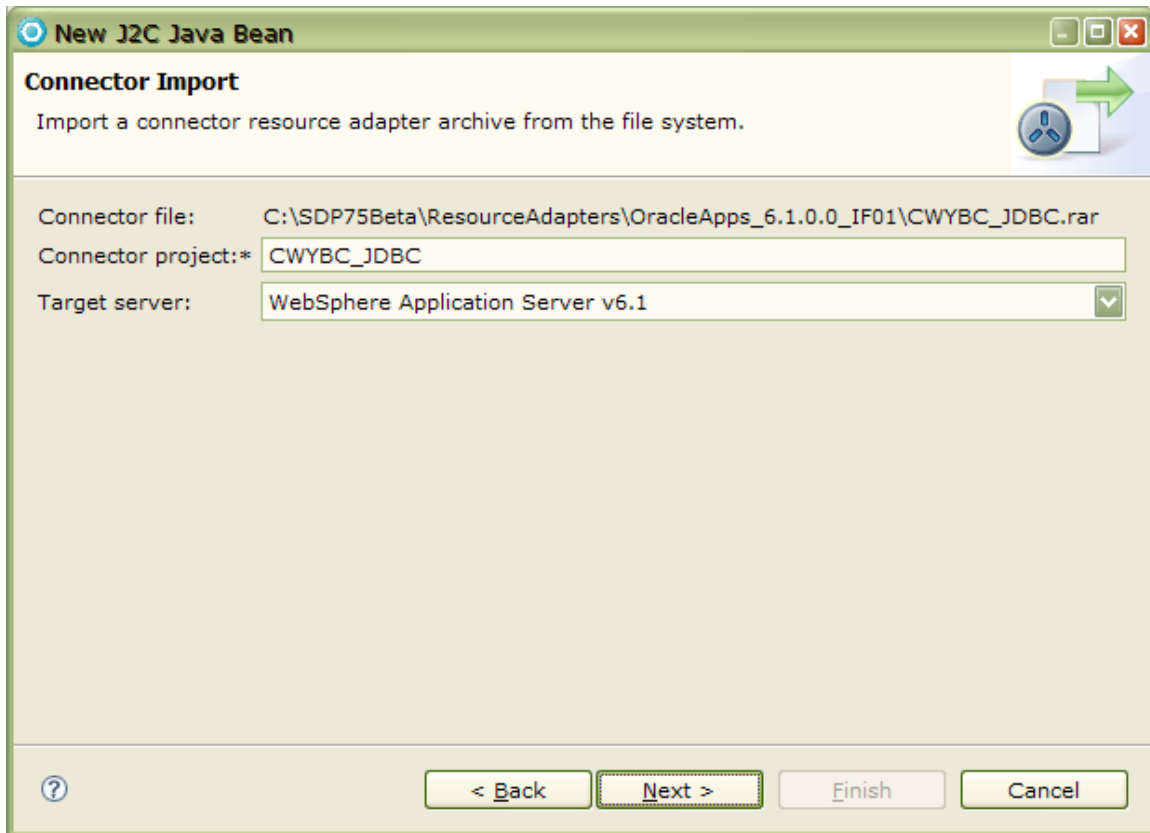
From the New menu select "Other", then in the New window scroll down to J2C entry and expand it. You should see the following:



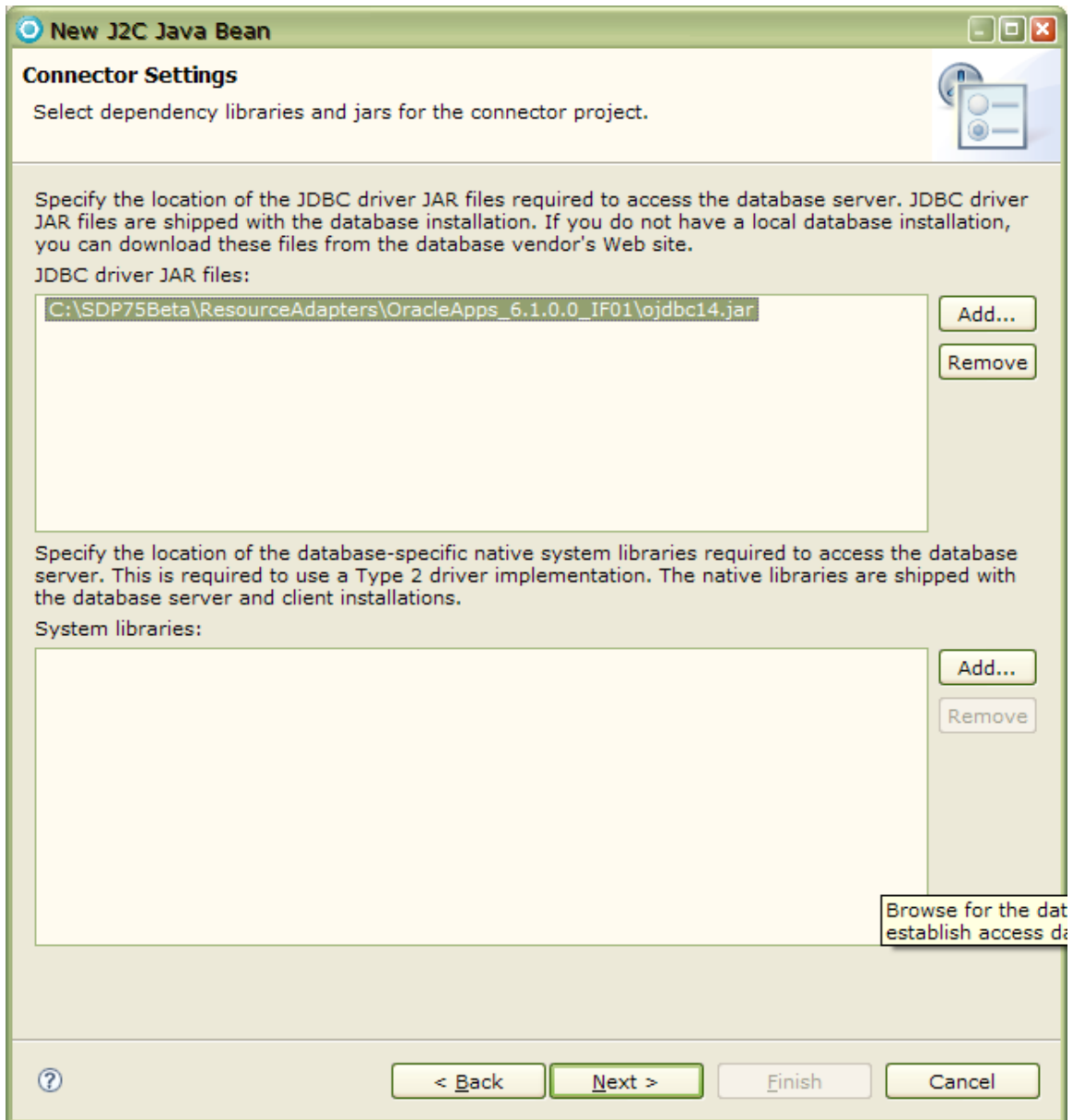
- Click button 'Next', we will see the following screenshot.



- Expand the entry for JDBC and select IBM WebSphere Adapter for JDBC. Click Next.
- In the New J2C Java Bean, Connector Import window, leave the Connector Project with the default value, and choose your server for the Target Server. Click Next.



- In the New J2C Java Bean, Connector Settings window, you need to select your JDBC driver files. To do this, click the Add button next to the window for JDBC driver JAR files and select your JDBC driver. Click Next.



- Please click button 'Next'

**New J2C Java Bean**

**Discovery Configuration**

The wizard can now guide you through discovery of objects to communicate with. Specify settings to begin discovery.

**Connection Configuration**

Database system connection information

Left Panel (Tree View):

- DB2 UDB
  - DB2 UDB iSeries
  - DB2 UDB zSeries
- Oracle
  - 8
  - 9
  - 10
- SQL Server
- Generic JDBC

Right Panel (Properties):

JDBC driver type: Oracle Thin Driver

Database: \* my\_database

Host name: \* my\_host

Port number: \* 1521

JDBC driver classname: \* oracle.jdbc.driver.OracleDr

Database URL: jdbc:oracle:thin:@my\_host

Additional JDBC driver connection properties [name=value;name=value]:

User name: \* user

Password: \* \*\*\*\*

Prefix for business object names:

Advanced >>

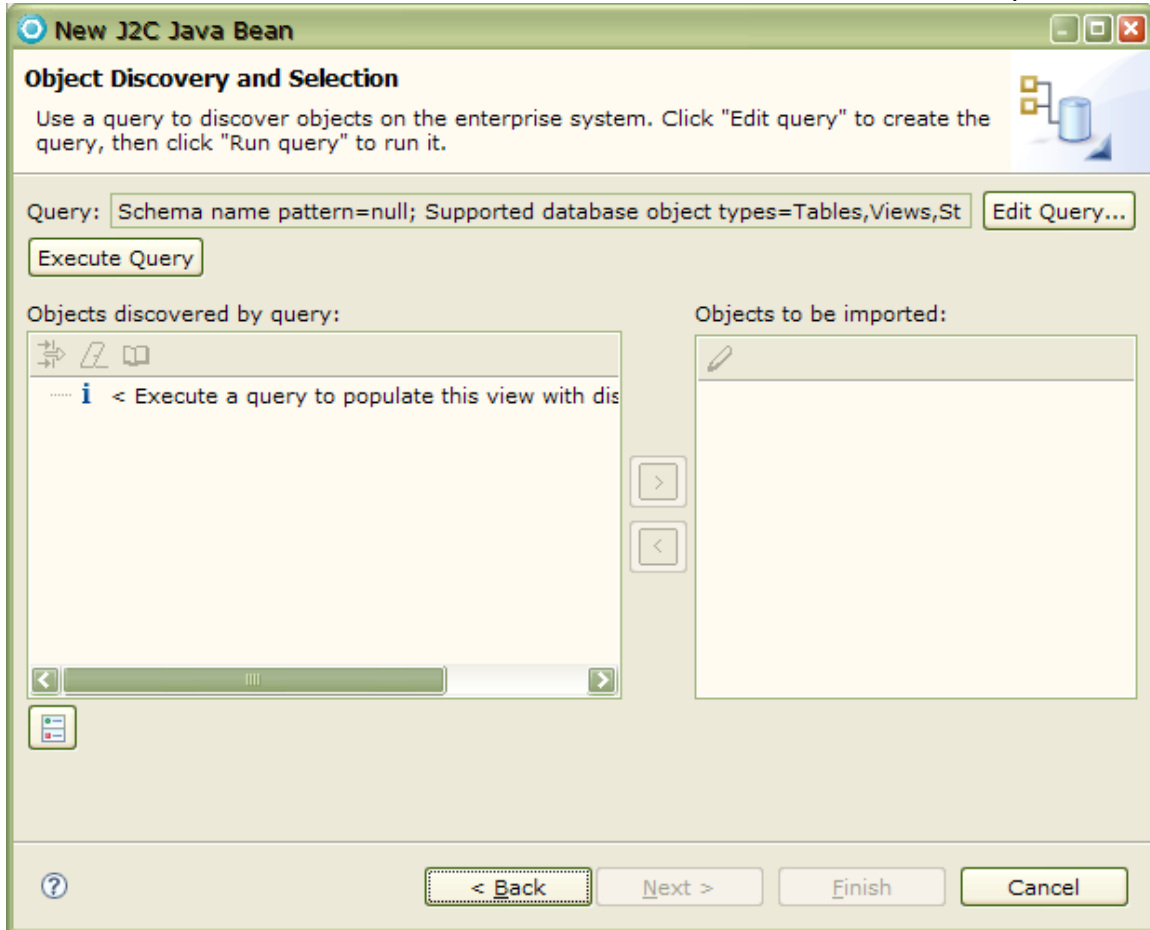
☐ Specify the level of the logging desired

Buttons: < Back, Next >, Finish, Cancel

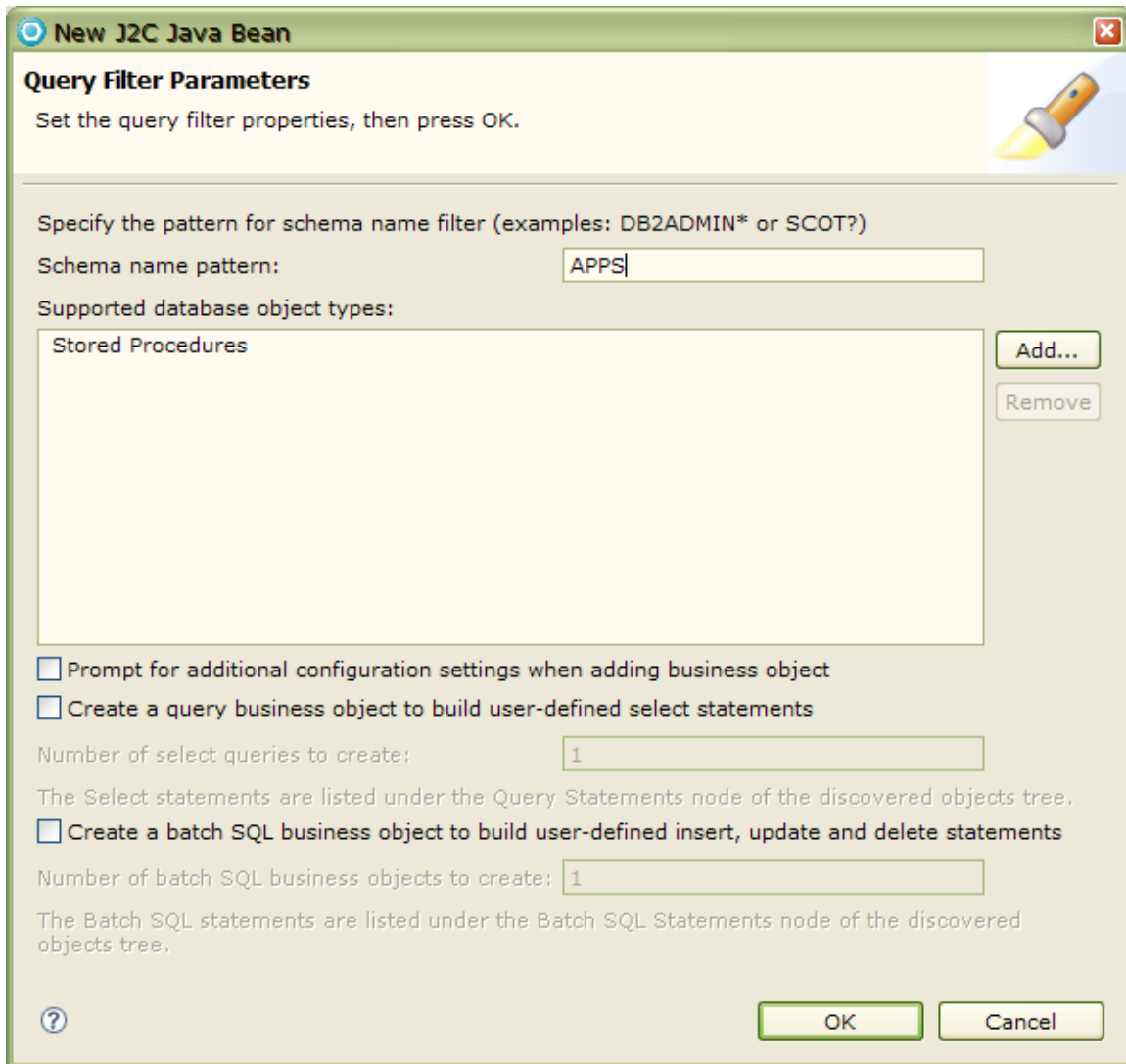
- Use the left panel to select the Database vendor, driver, and version that the adapter will be connecting to.
- On the right panel fill out the information below (some of the information will be provided by default):
  1. Database Name
  2. Database Host
  3. Database Port
  4. JdbcDriverClass
  5. Database URL
  6. UserName
  7. Password
- Click Next.

## Selecting the objects for the Oracle API

- Before you select the object, you need to edit the query that will run.
  - In the Find and Discover Services screen click the Edit Query button.



- Enter 'APPS' in the Schema Name Filter field.
- Remove Tables.
- Remove Views.
- Remove Synonyms.



The image shows a 'New J2C Java Bean' dialog box with a title bar containing a question mark icon and a close button. The main area is titled 'Query Filter Parameters' and includes a pencil icon. It contains several sections: a text box for 'Schema name pattern' with 'APPS' entered; a list box for 'Supported database object types' with 'Stored Procedures' selected and 'Add...'/'Remove' buttons; two unchecked checkboxes for additional configuration; a text box for 'Number of select queries to create' with '1' entered; a text box for 'Number of batch SQL business objects to create' with '1' entered; and 'OK'/'Cancel' buttons at the bottom.

**New J2C Java Bean**

**Query Filter Parameters**  
Set the query filter properties, then press OK.

Specify the pattern for schema name filter (examples: DB2ADMIN\* or SCOT?)

Schema name pattern:

Supported database object types:

☐ Prompt for additional configuration settings when adding business object

☐ Create a query business object to build user-defined select statements

Number of select queries to create:

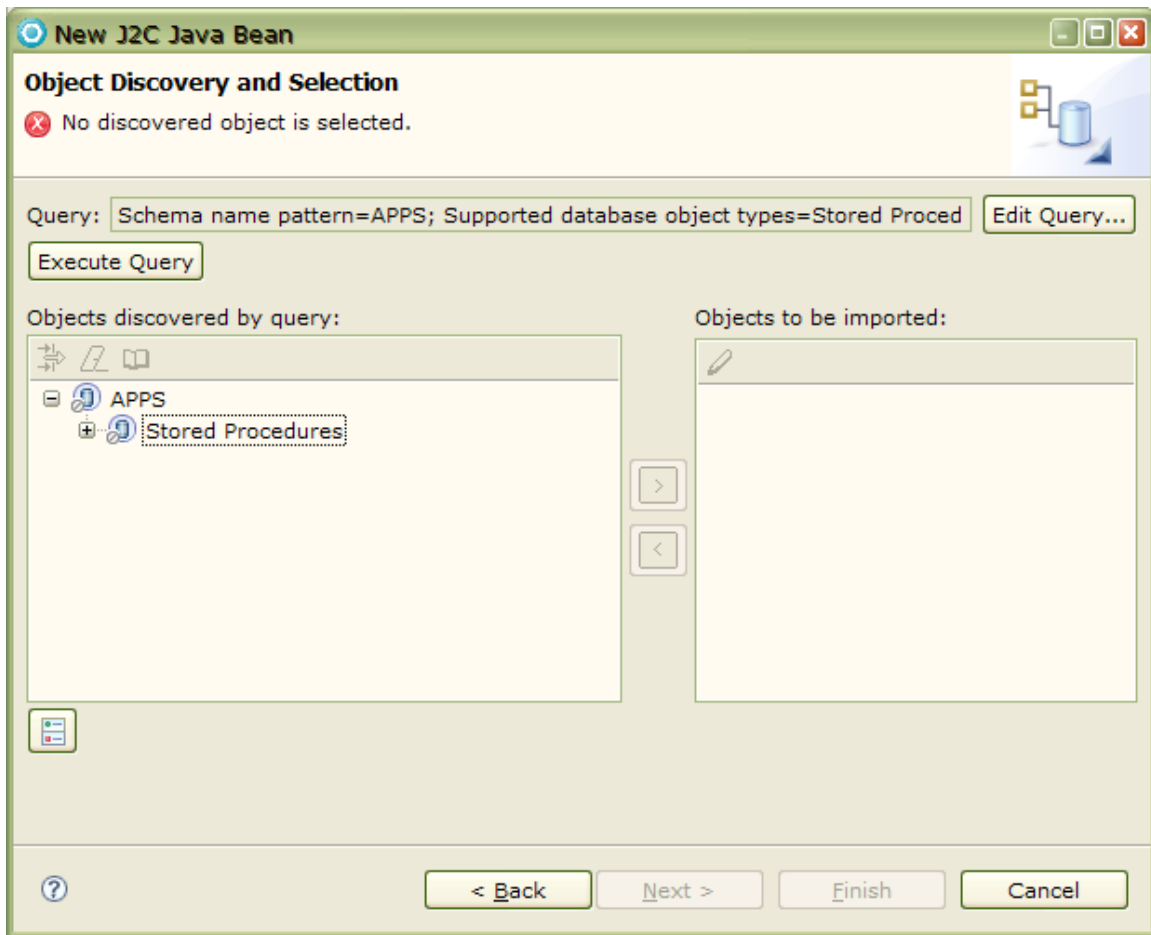
The Select statements are listed under the Query Statements node of the discovered objects tree.


☐ Create a batch SQL business object to build user-defined insert, update and delete statements

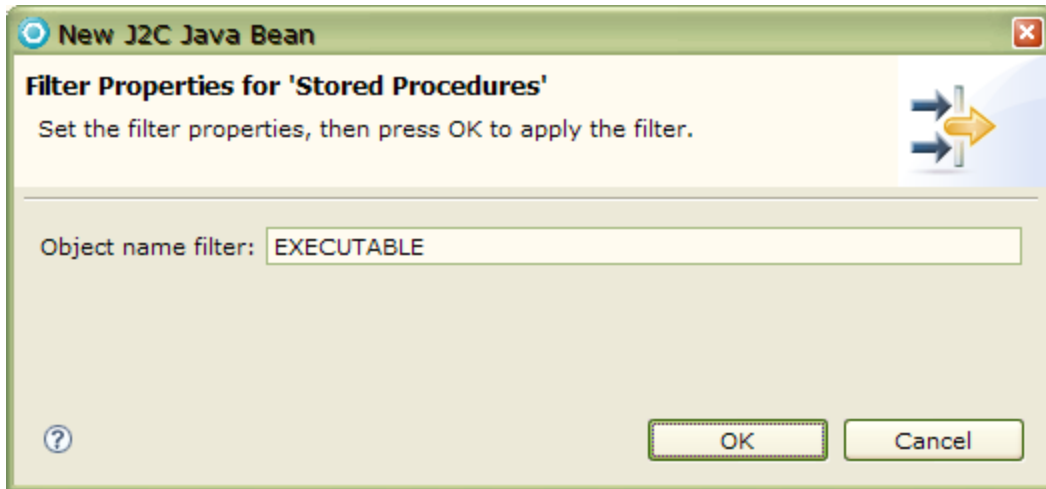
Number of batch SQL business objects to create:

The Batch SQL statements are listed under the Batch SQL Statements node of the discovered objects tree.

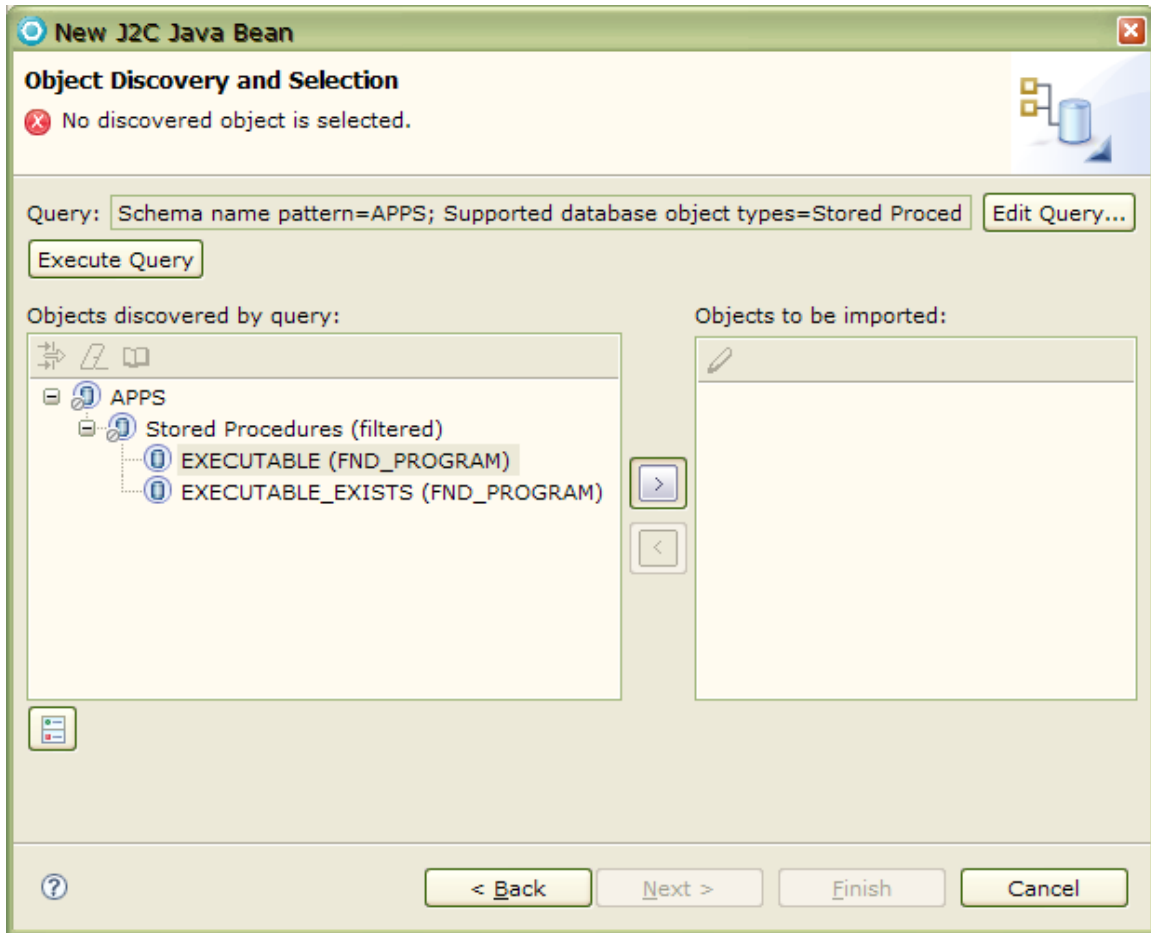
- Click OK.
- Click the Execute Query button.
- Expand the APPS schema.



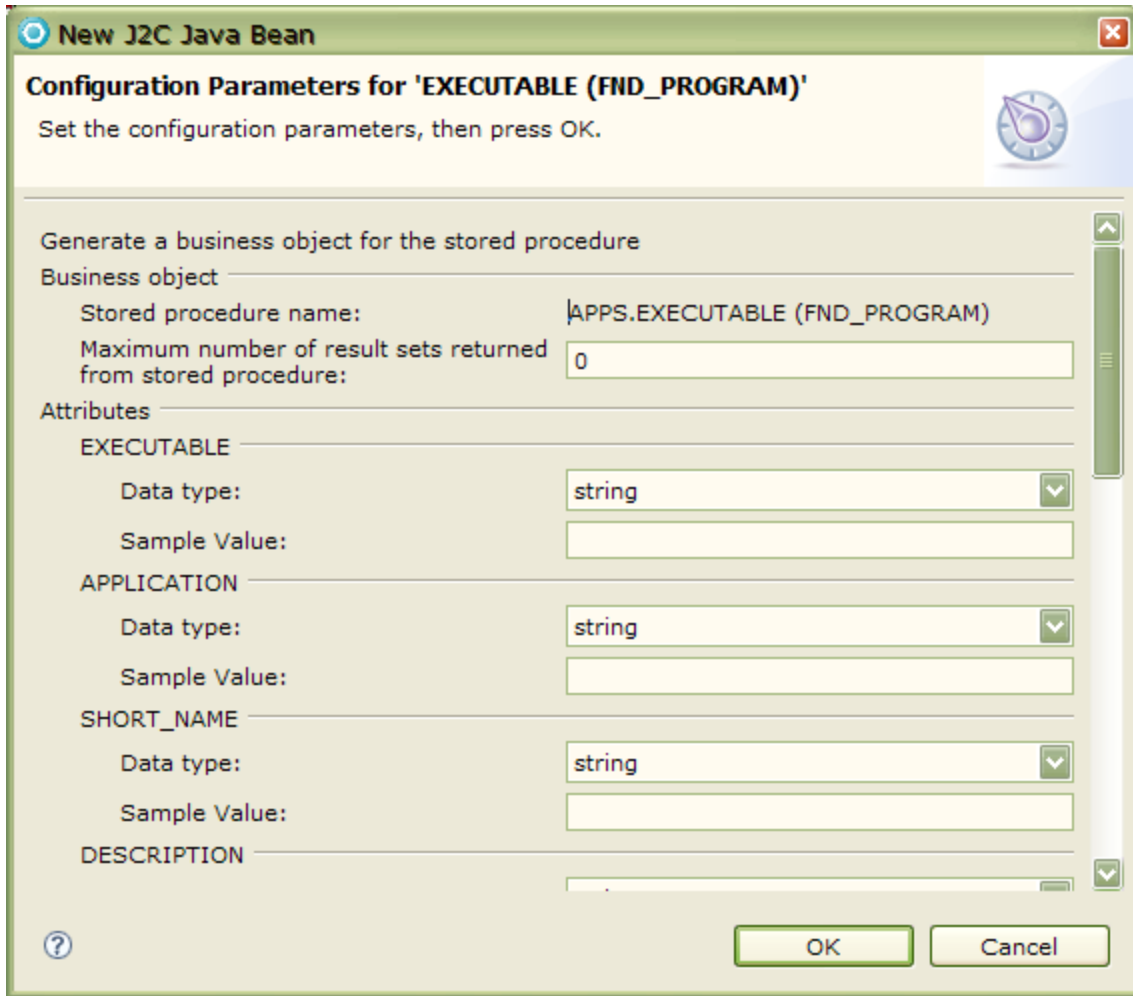
- Click on the word 'Stored Procedures' (**Don't expand them yet.**)
- Click the Filter button .
- In the Object Name Filter in the Filter Properties window, type: EXECUTABLE



- In the Object Name Filter in the Filter Properties window, type: EXECUTABLE
- Click OK.
- Expand the 'Stored Procedures' subcategory. (NOTE: Due to the large number of procedures contained in the APPS schema, it may take a long time for the adapter to return the list of Stored Procedures.)



- Select EXECUTABLE(FND\_PROGRAM) from the list of stored procedures
- Click Add (>). A window titled 'Configuration Parameters for EXECUTABLE(FND\_PROGRAM)' will appear.



The image shows a 'New J2C Java Bean' configuration window. The title bar says 'New J2C Java Bean'. The main heading is 'Configuration Parameters for 'EXECUTABLE (FND\_PROGRAM)'. Below this, it says 'Set the configuration parameters, then press OK.' There is a 'Generate a business object for the stored procedure' checkbox which is checked. Under 'Business object', 'Stored procedure name' is 'APPS.EXECUTABLE (FND\_PROGRAM)' and 'Maximum number of result sets returned from stored procedure' is '0'. Under 'Attributes', there are three sections: 'EXECUTABLE', 'APPLICATION', and 'SHORT\_NAME'. Each has a 'Data type' dropdown set to 'string' and a 'Sample Value' text box. There is also a 'DESCRIPTION' text box at the bottom. The window has 'OK' and 'Cancel' buttons at the bottom right.

**New J2C Java Bean**

**Configuration Parameters for 'EXECUTABLE (FND\_PROGRAM)'**

Set the configuration parameters, then press OK.

☒ Generate a business object for the stored procedure

**Business object**

Stored procedure name:

Maximum number of result sets returned from stored procedure:

**Attributes**

**EXECUTABLE**

Data type:

Sample Value:

**APPLICATION**

Data type:

Sample Value:

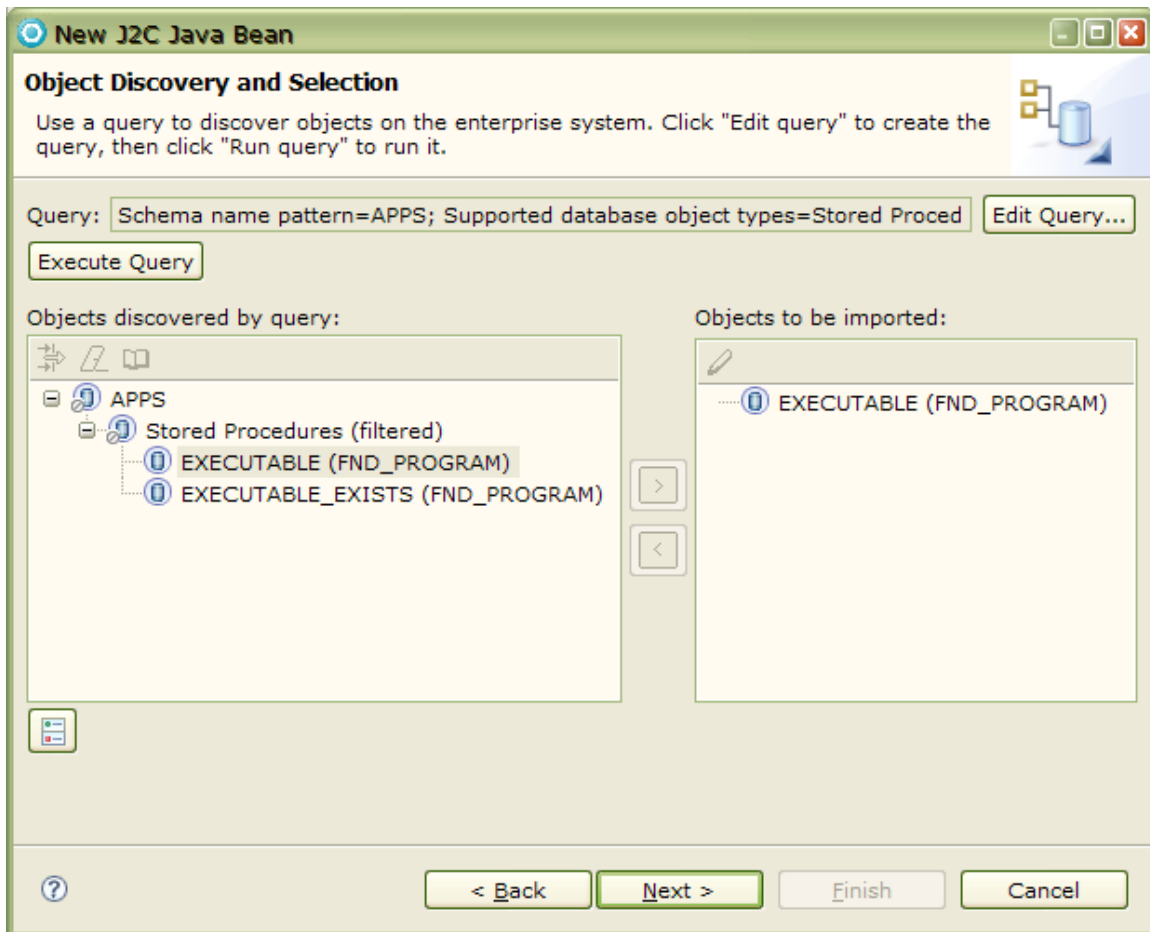
**SHORT\_NAME**

Data type:

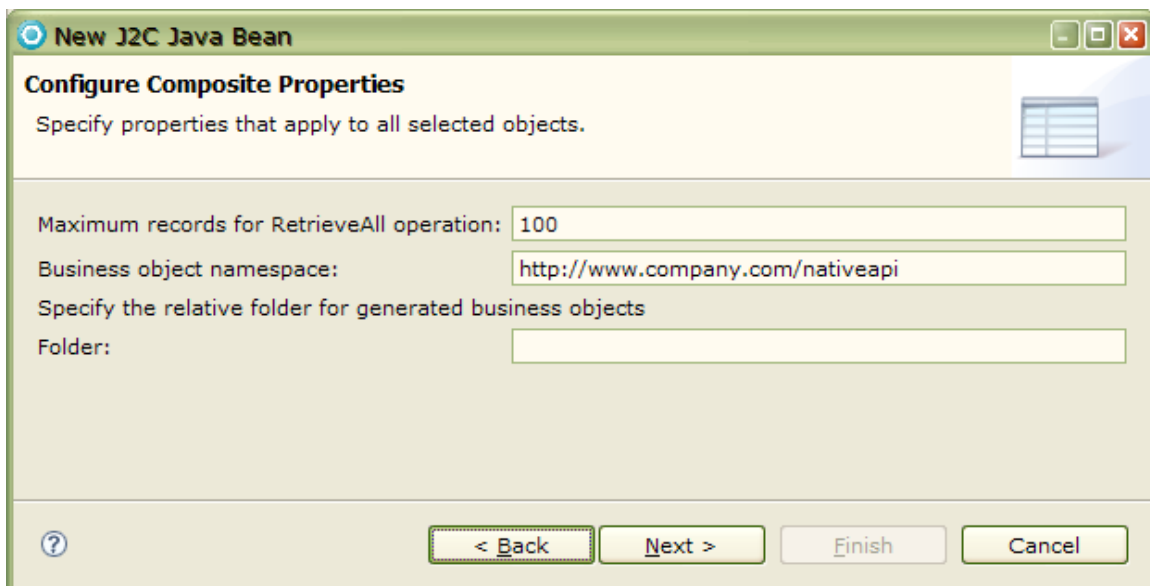
Sample Value:

**DESCRIPTION**

- Click OK. The Stored Procedure will be added to the 'Objects to be imported:' box.



- Click Next.



- Type “http://www.company.com/naiveapi” in the Business object namespace field. This namespace will be generated in the XSDs representing business data and used to generate Java data bindings.
- Press “Next”

**New J2C Java Bean**

**Java Bean Creation and Deployment Configuration**

✖ Project Name: cannot be empty.

**Save properties**

Project Name: \*  Browse... New

Package Name:  Browse... New

Interface Name:

Implementation Name:

**Generate Command Bean**

Generate Command Bean for method: executeAppsFnd\_ProgramU46executable

☐ Enable generate Command Bean

executeAppsFnd\_ProgramU46executable:

Command Bean name:

Command Bean input name:

Command Bean output name:

**Connection properties**

Managed connections are obtained through JNDI name lookup from the application server.  
Non-managed connections are obtained directly from the resource adapter.

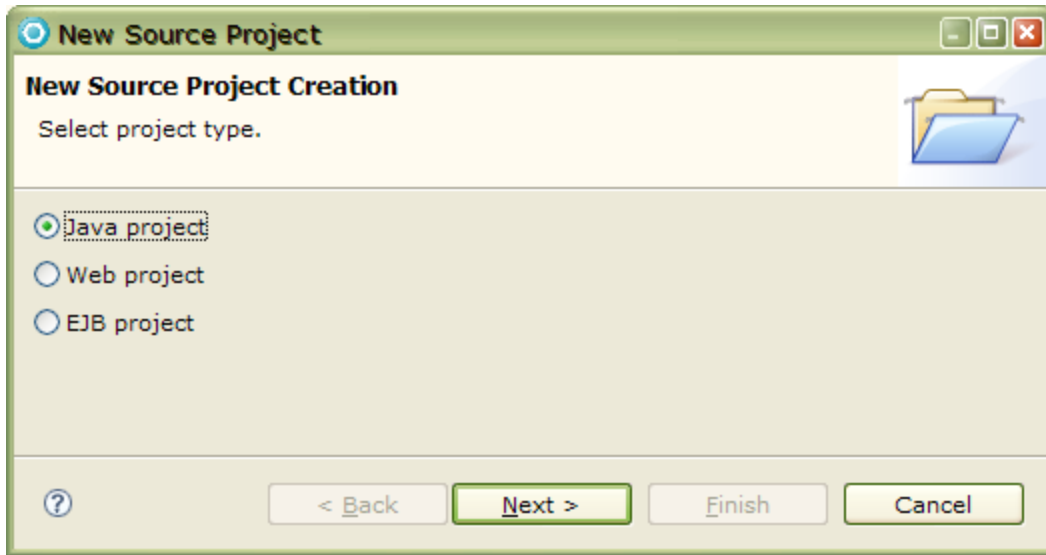
☒ Managed Connection (recommended)

JNDI Lookup Name:  Browse... New

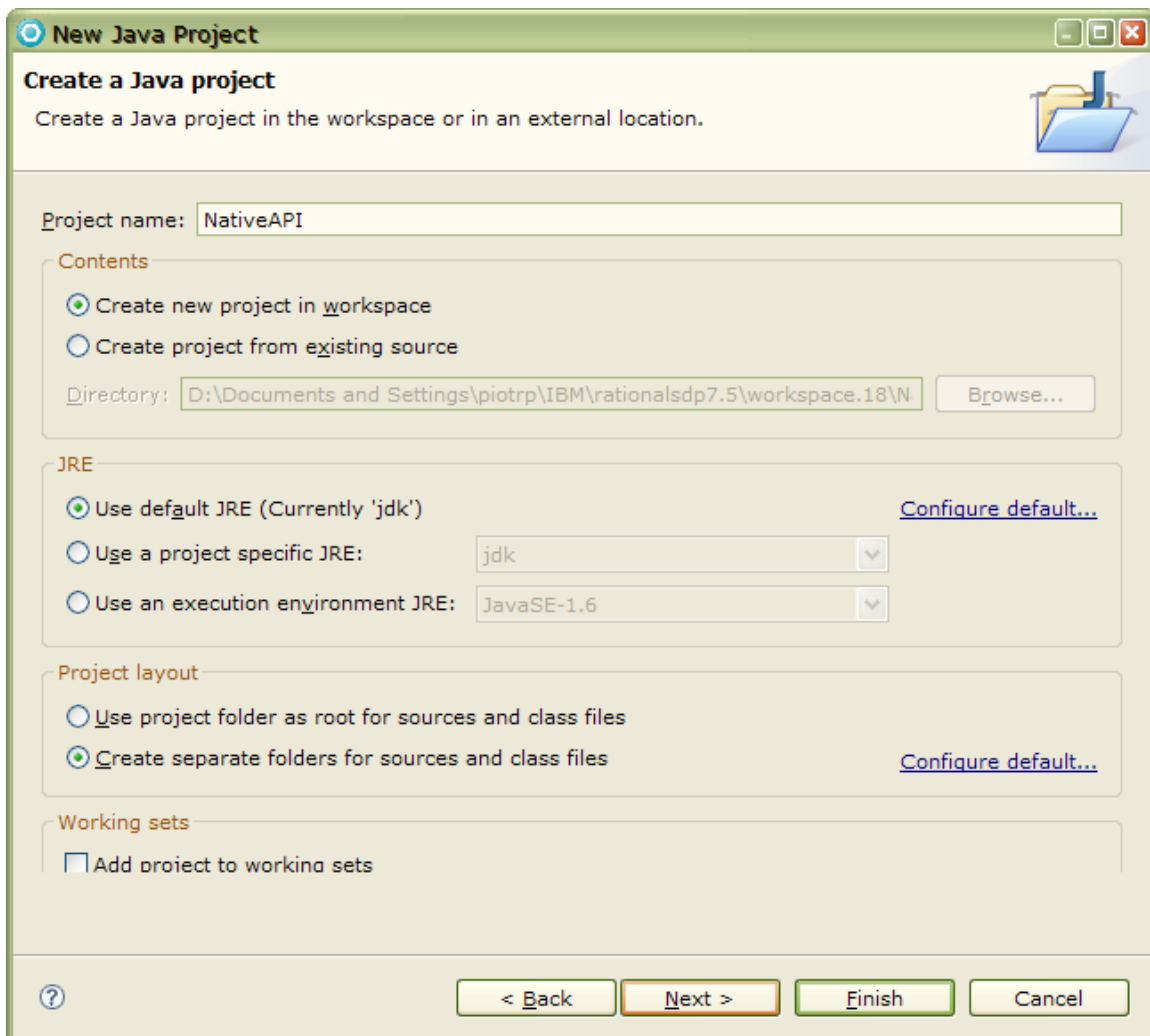
☐ Non-managed Connection

< Back Next > Finish Cancel

- Press “New” to create new project to contain the application.



- Select Java Project and click Next.



- Specify Project Name, “NativeAPI” and Click Finish
- In the Publishing Properties, create new package, “nativeapi.executable” and the interface name “EBSAPI”.
- The implementation name will be filled automatically.

**New J2C Java Bean**

**Java Bean Creation and Deployment Configuration**

Specify the properties for creating and running the J2C Java bean.

**Save properties**

Project Name: \* NativeAPI Browse... New...

Package Name: \* nativeapi.executable Browse... New...

Interface Name: \* EBSAPI

Implementation Name: \* EBSAPIImpl

**Generate Command Bean**

Generate Command Bean for method: executeAppsFnd\_ProgramU46executable

☐ Enable generate Command Bean

executeAppsFnd\_ProgramU46executable:

Command Bean name:

Command Bean input name:

Command Bean output name:

**Connection properties**

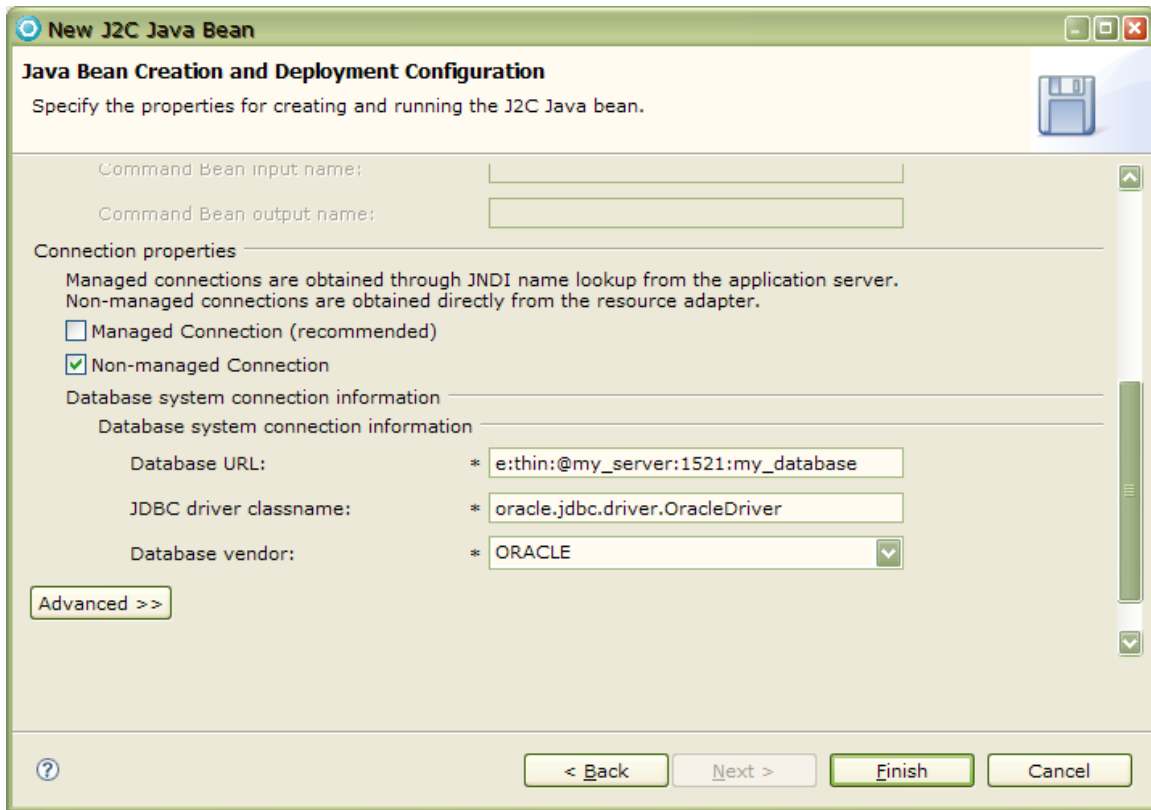
Managed connections are obtained through JNDI name lookup from the application server.  
Non-managed connections are obtained directly from the resource adapter.

☐ Managed Connection (recommended)

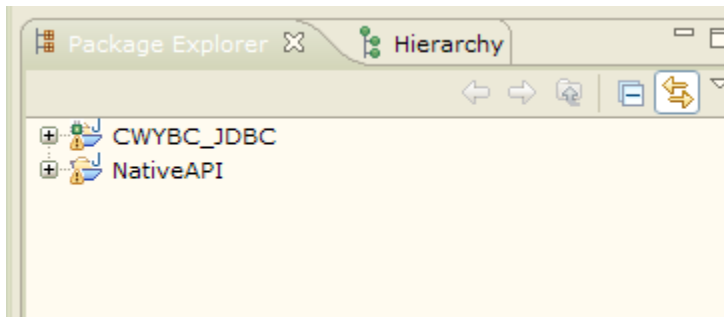
☒ Non-managed Connection

< Back Next > Finish Cancel

- Deselect the option Managed Connection
- Select Non-managed Connection

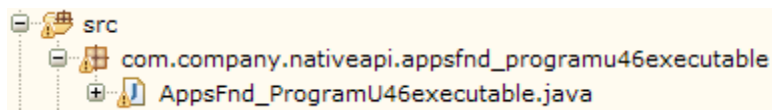


- The Database Connection Information will be prefilled.
- Click Finish
- The specified project will be created and you will see the following in the Project Explorer

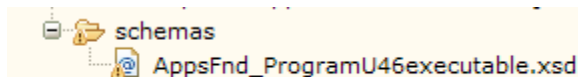


The set of generated artifacts includes

- Java Data Binding representing data exchanged with the Oracle EBS system



- The metadata represented as schema (AppFn\_d\_ProgramU46executable.xsd)



- J2C Java Bean, interface and implementation: EBSAPI, EBSAPIImpl



The last part of recreating the PeopleSoft sample requires creation of the non-managed driver program to instantiate and interact with the generated J2C Java Bean. Using RAD tools create package named “client” in the NativeAPI project. Within this package, create class “ApplicationClient”.



Paste the following code snippet as ApplicationClient class content:

```
package client;

import javax.resource.ResourceException;
import nativeapi.executable.EBSAPI;
import nativeapi.executable.EBSAPIImpl;
```

```

import
com.company.nativeapi.appsfn_d_programu46executable.AppsFn_d_ProgramU46ex
ecutable;

public class ApplicationClient {

    public static void main(String[] args) {

        try {
            // Create input instance of java data binding
            AppsFn_d_ProgramU46executable inputData = new
AppsFn_d_ProgramU46executable();

            // Create output instance of java data binding
            AppsFn_d_ProgramU46executable outputData = null;

            // Populate input data
            inputData.setapplication("AR");
            inputData.setexecutable("IBMSAMPLENAME1");

            inputData.setexecution_file_name("TEST_PACKAGE.TEST_PROCEDURE");
            inputData.setexecution_method("PL/SQL Stored Procedure");
            inputData.setlanguage_code("US");
            inputData.setshort_name("IBMSAMPLENAME1");

            // Create instance of the J2C Java Bean
            EBSAPI storedProcedure = new EBSAPIImpl();

            // Invoke executable program using adapter
            outputData =
storedProcedure.executeAppsFn_d_ProgramU46executable(inputData);

            // Print returned subroutine name
            if(outputData != null)
                System.out.println(outputData.getsubroutine_name());
        }
        catch (ResourceException e) {
            System.out.println("Exception during execution: " +
e.getMessage());
        }
    }
}

```

### Verify the results using the Oracle EBS web client

- Log into Oracle EBS.
- Select System Administrator responsibility.
- Select Concurrent -> Program option. This will bring up a Concurrent Program screen.
- Query the window for the executable.
- In the Short Name field, type 'IBMSAMPLENAME'. Click Go.
- You should see the executable details listed.

## **Clearing the sample content**

The only new artifact introduced to Oracle is the executable program. To remove it, use the same steps above for verifying the content. After the executable is located, choose the menu bar option Edit, Delete, and save your changes.

## Notices

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

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

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

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

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

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

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

sites is at your own risk. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Software Interoperability Coordinator, Department 49XA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

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

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

#### **COPYRIGHT LICENSE:**

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

### **Programming interface information**

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

**Warning:** Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

## Trademarks

IBM and related trademarks: <http://www.ibm.com/legal/copytrade.shtml>

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

This product includes software developed by the Eclipse Project (<http://www.eclipse.org>).