



How to Create an Online Mobile Application with SAP Web Dynpro

In this tutorial, an online application using SAP Web Dynpro is created.

First, a simple web application is created and shown on a browser, either on a laptop or a mobile device.

This application is then extended so that it connects to a backend SAP R/3 system.

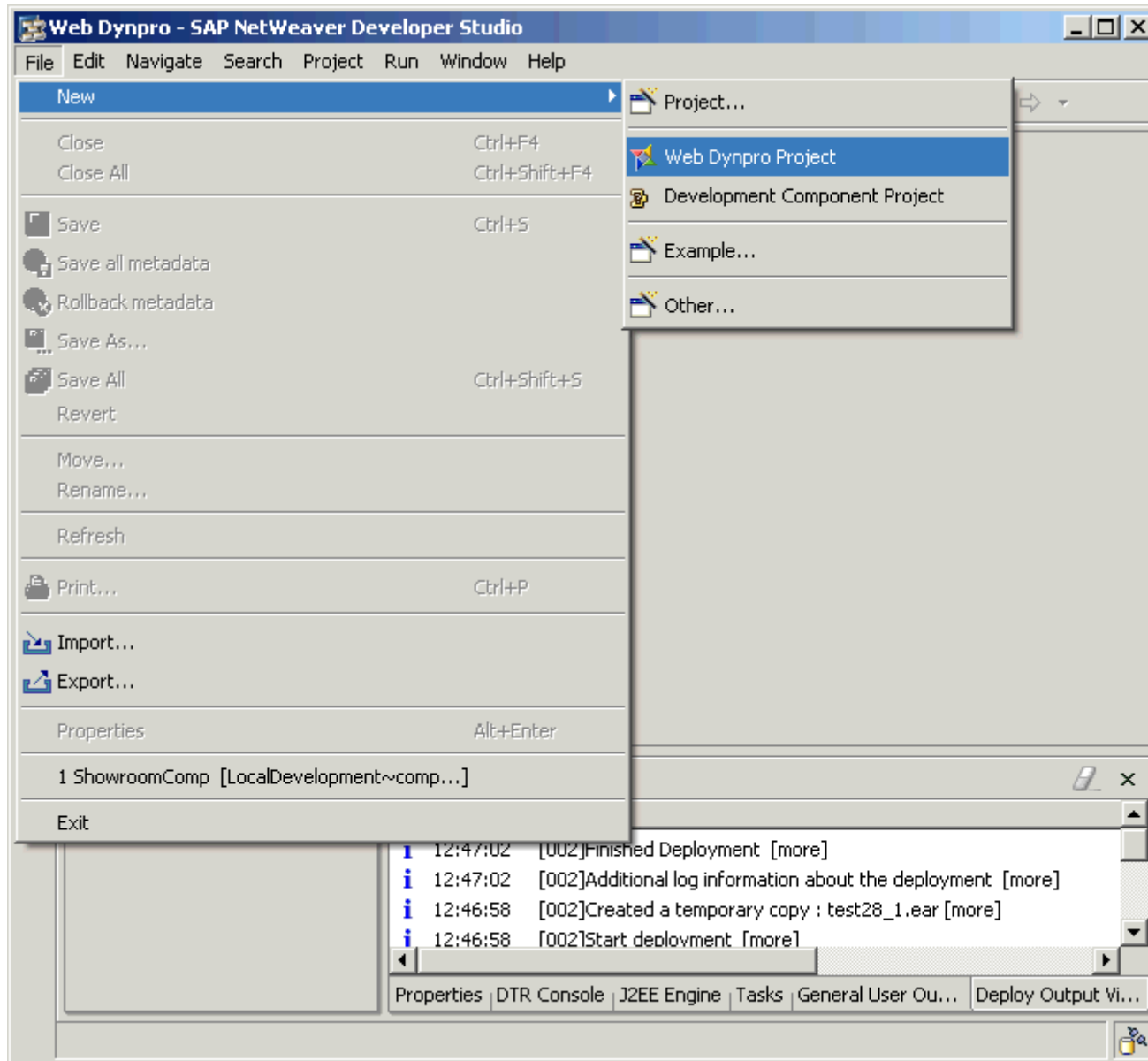
The application compiles a list of contact names from partner organisations. This is displayed in the top part of the browser.

On clicking on a partner or a contact name, the contact details are shown in the bottom half of the table.

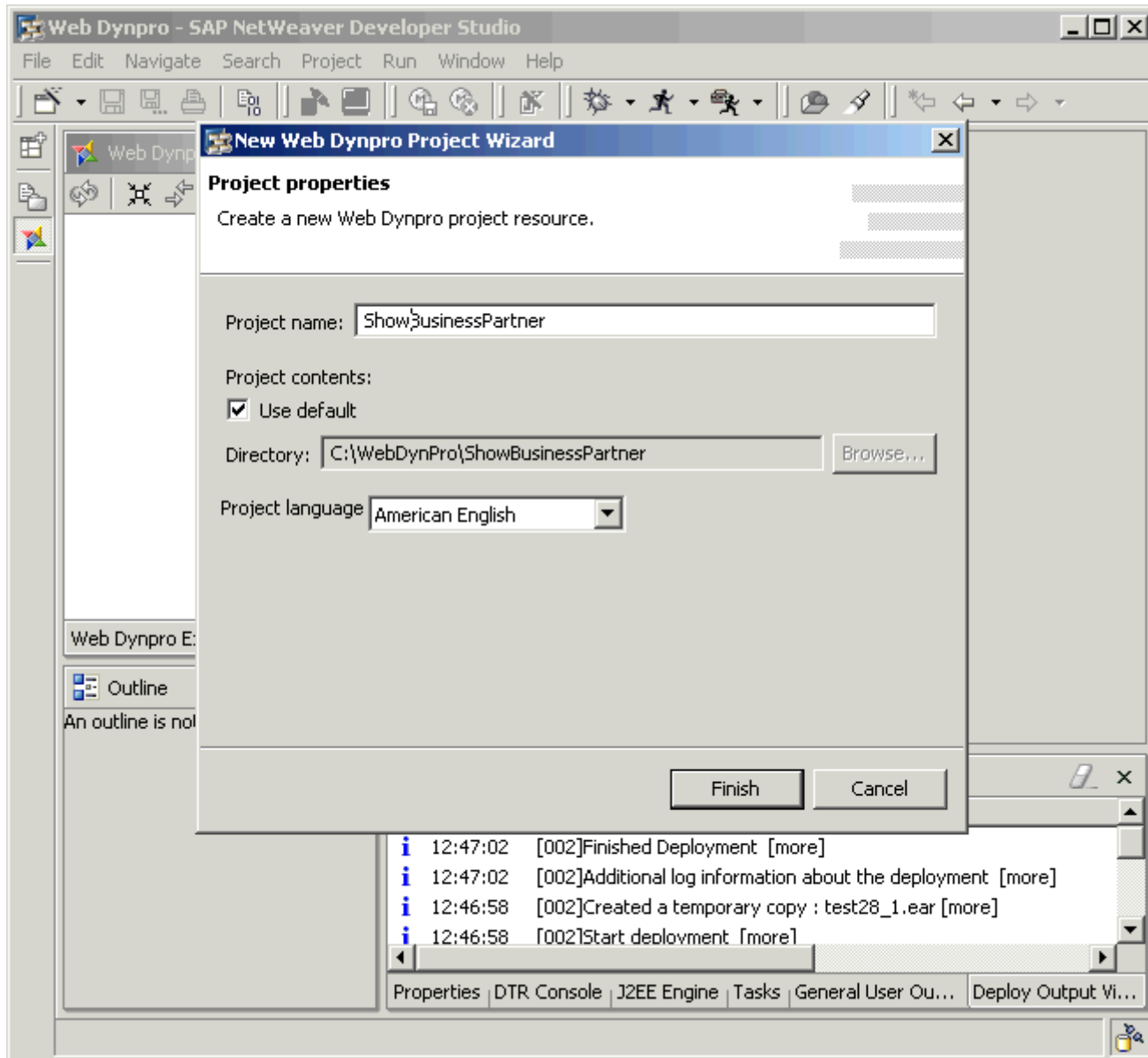
Let's start with the first part of the application

.... a simple web application.

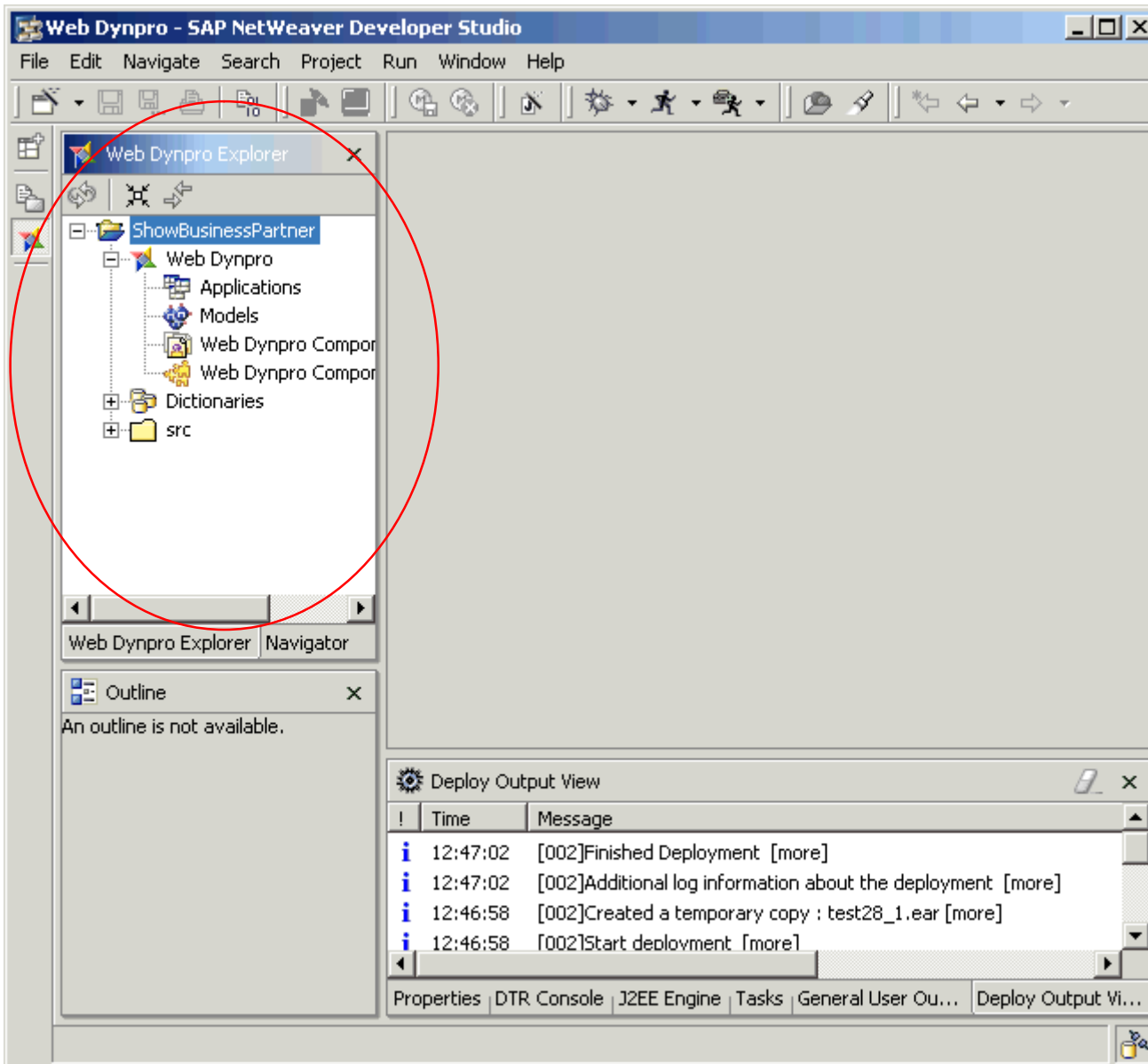
Simple Web Application



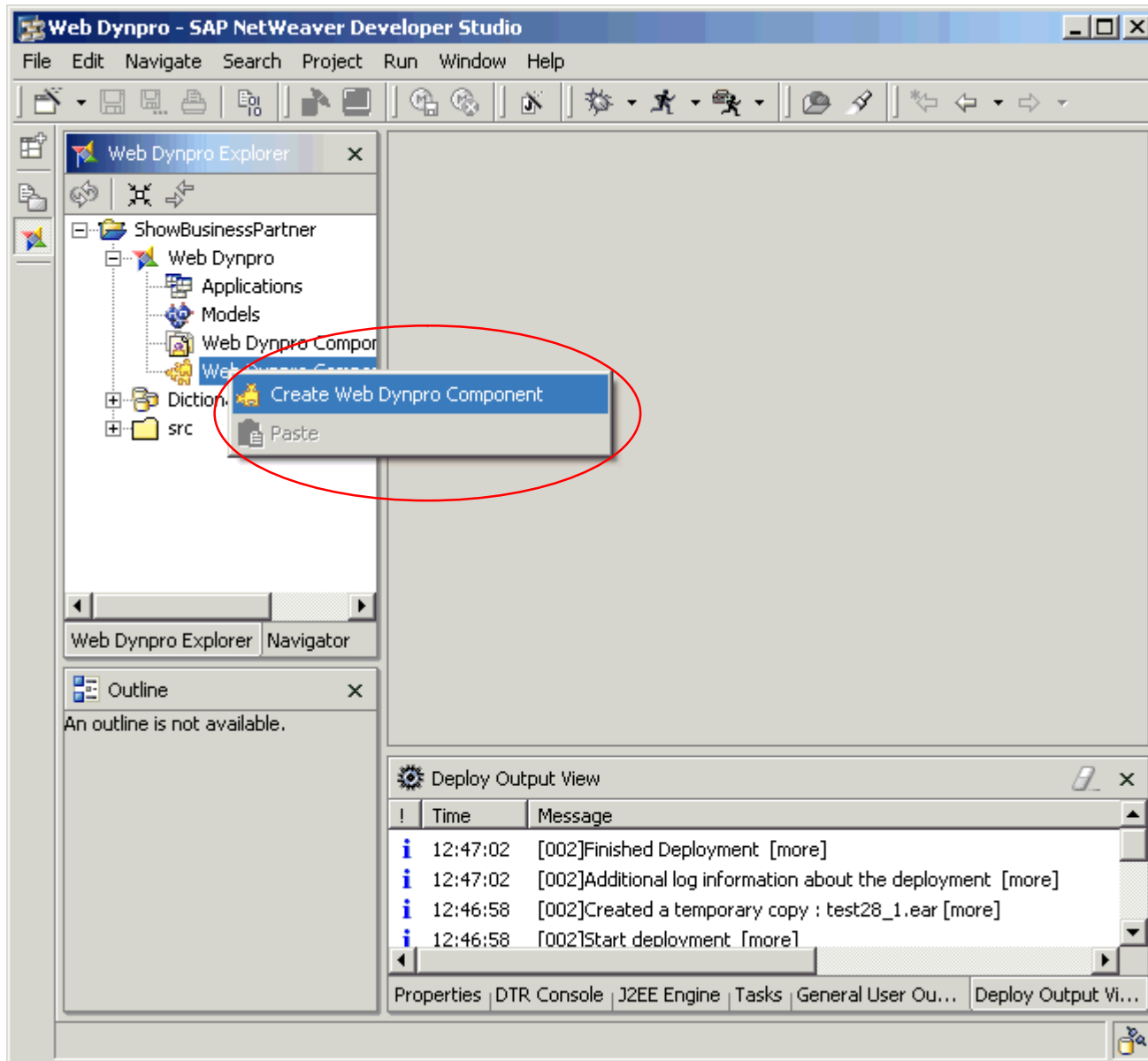
Create a new Web Dynpro Project first.



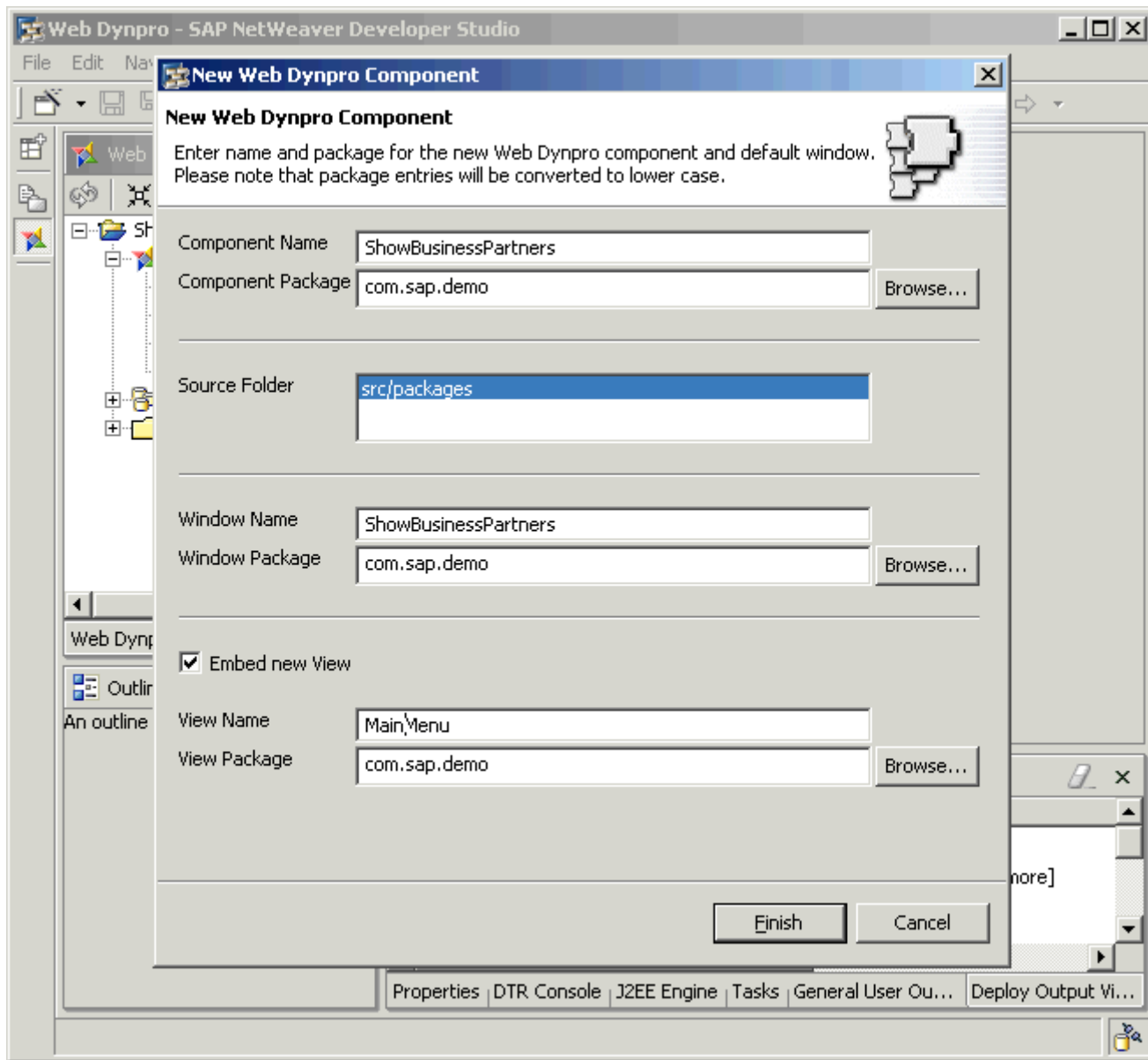
and give it a name.



**Web Dynpro
creates the project
and some new
nodes in the
project explorer.**

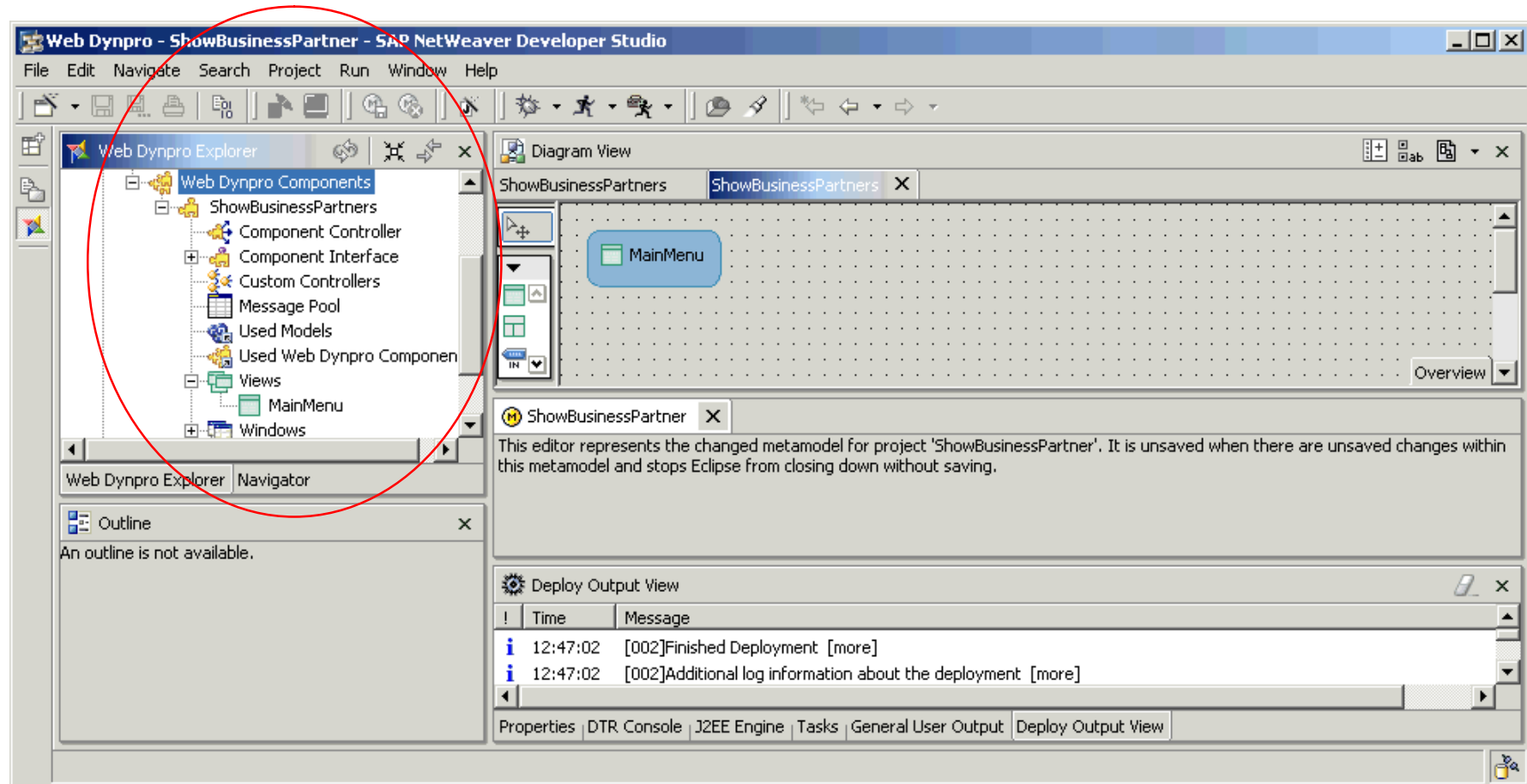


Create a new component with a right mouse click.

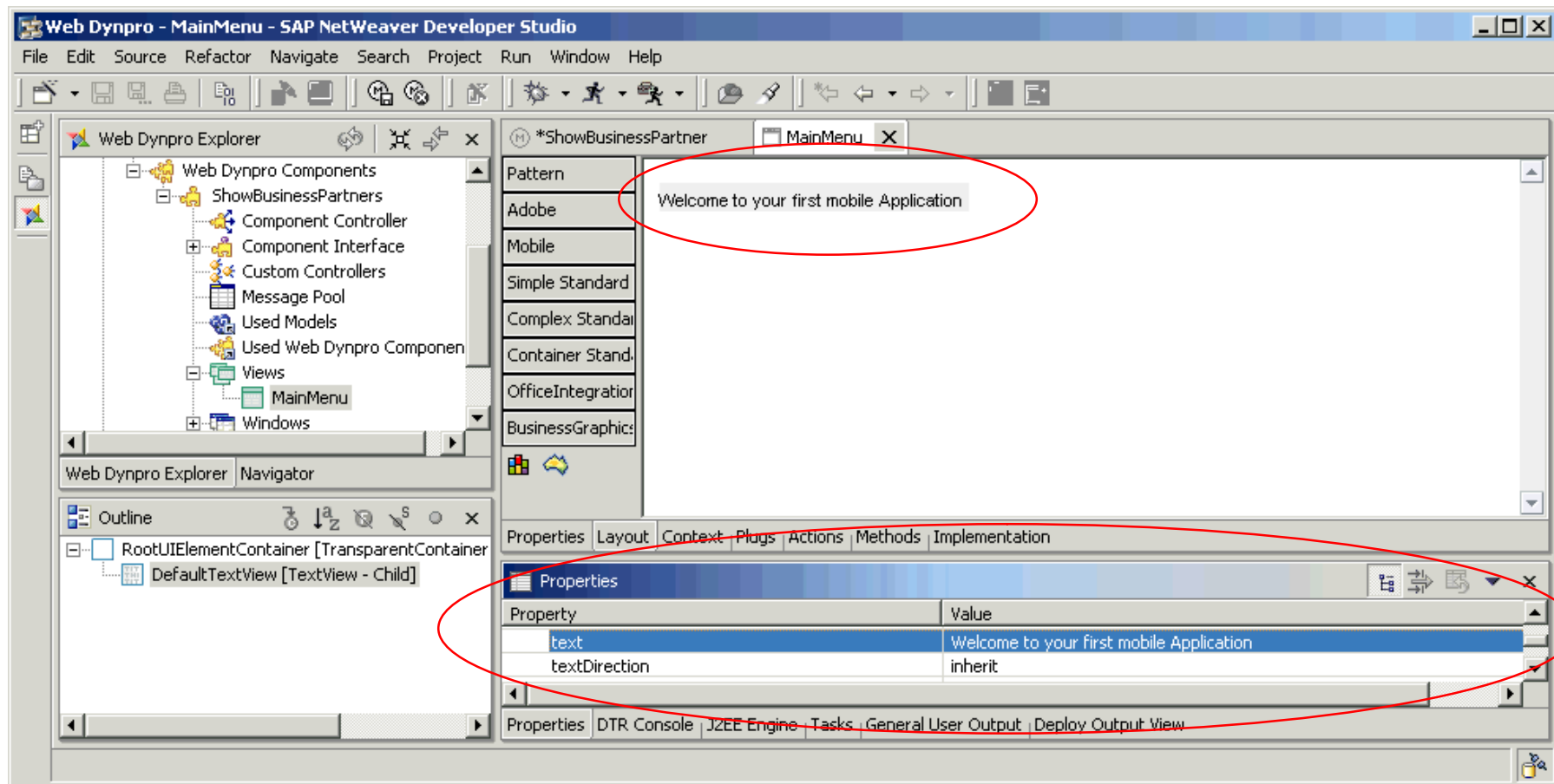


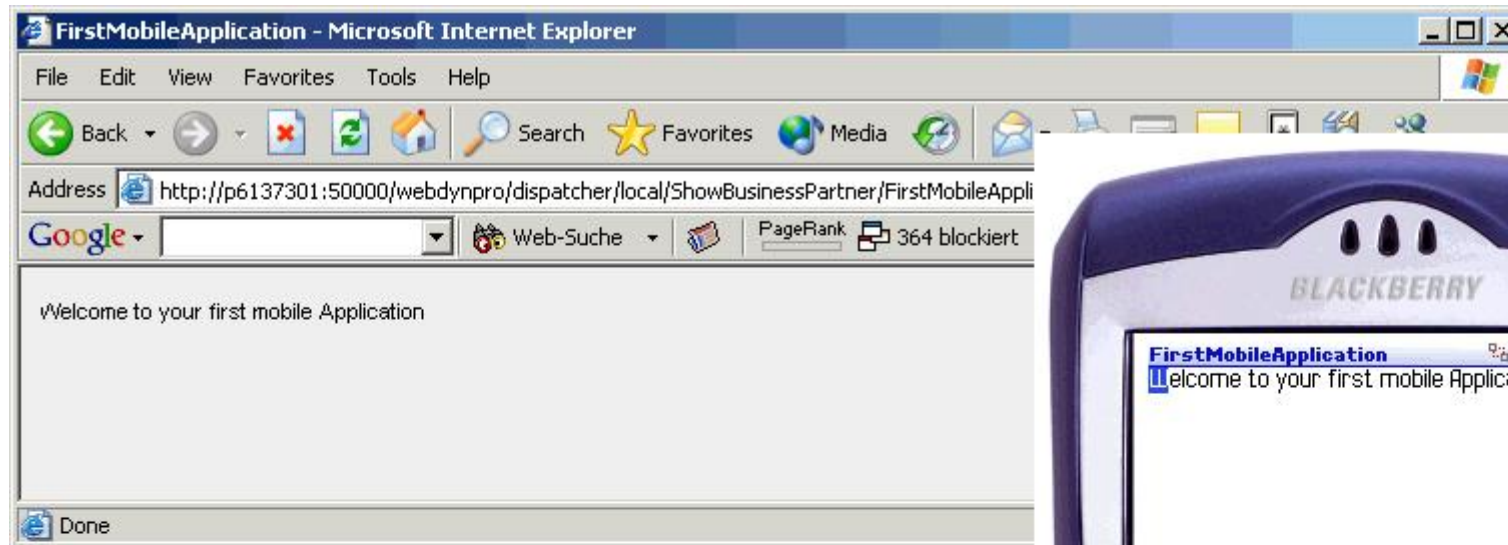
and give it a name.

Web Dynpro automatically creates some more nodes and a default view.



A default View was created with a UI element (user Interface element) showing a title. The text can be edited in the lower properties panel.





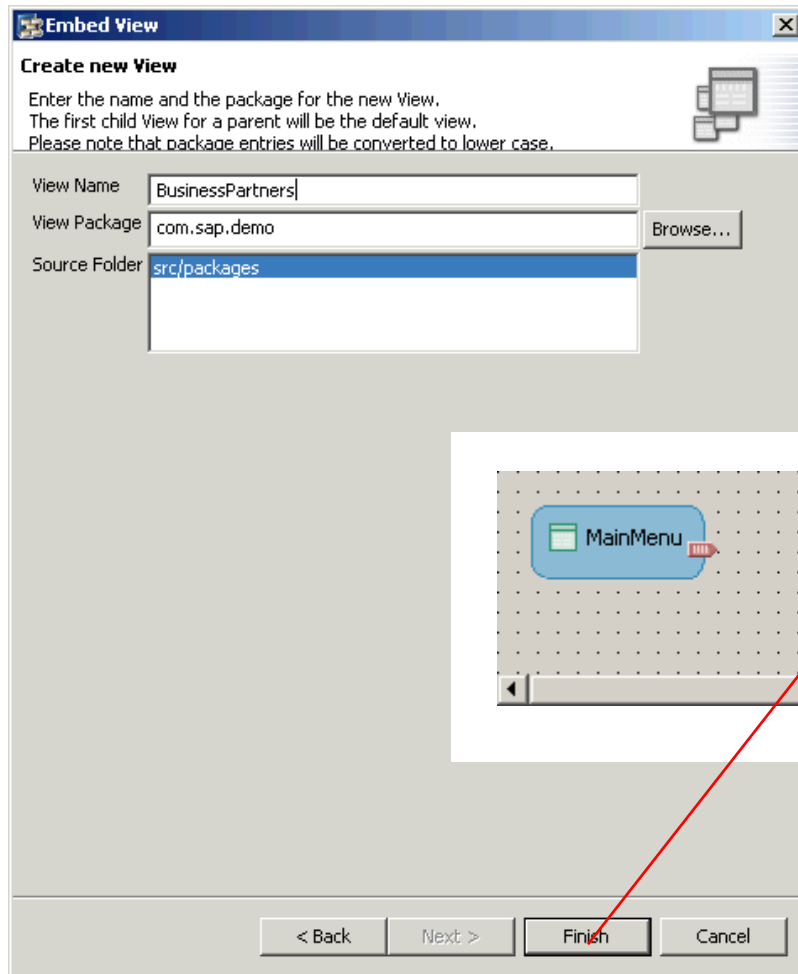
With a right mouse click on the application node. The application is deployed to the SAP Web Application Server. A PC browser or a simulator can be used to test the application.

Copy the url into a mobile device browser to see your first mobile application.

You have just created your first Web Dynpro application.

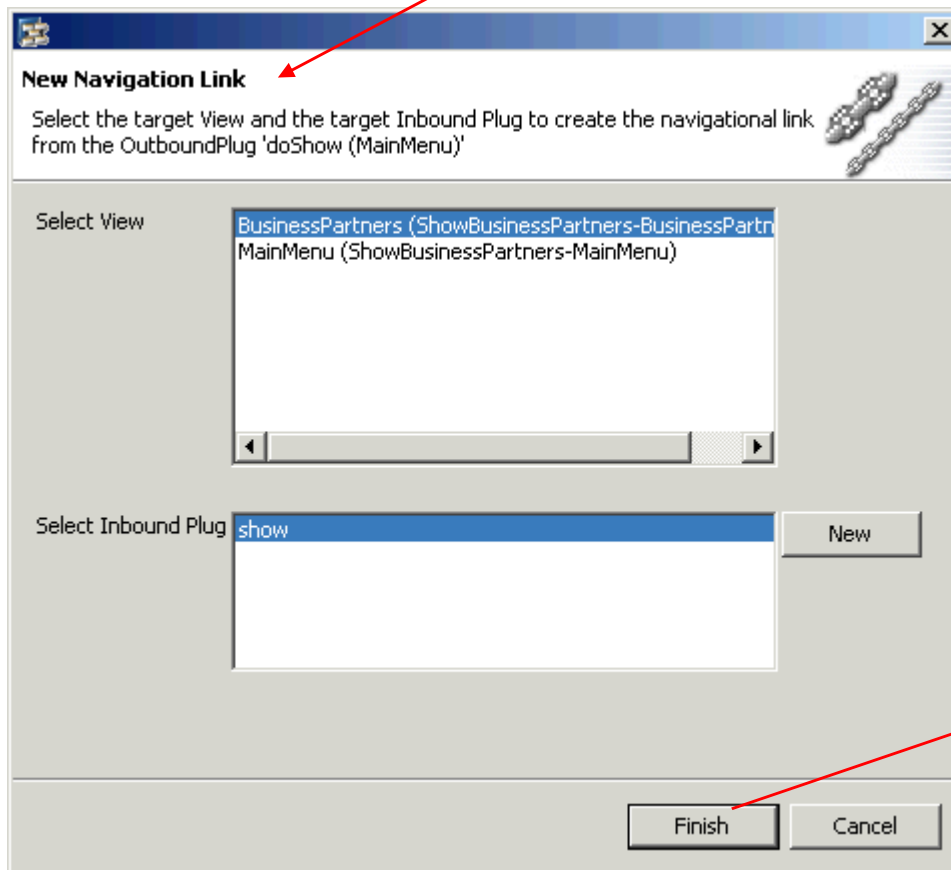
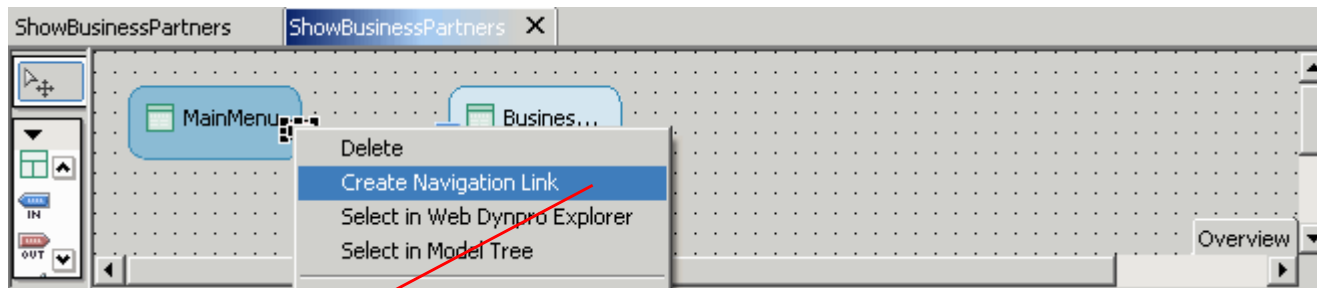


Extend the application to connect to backend SAP R/3



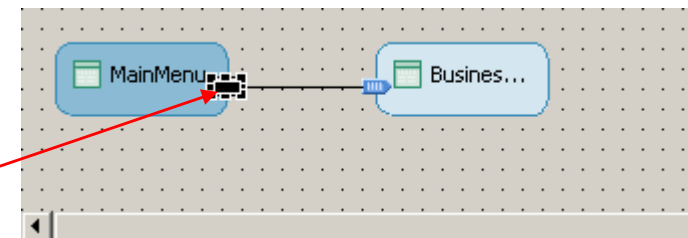
You can develop the application you have just created to access data from a backend SAP R/3 system

Start by adding a new View. Add an Outbound Plug to the view of the main menu and an Inbound Plug to the new view using drag and drop.

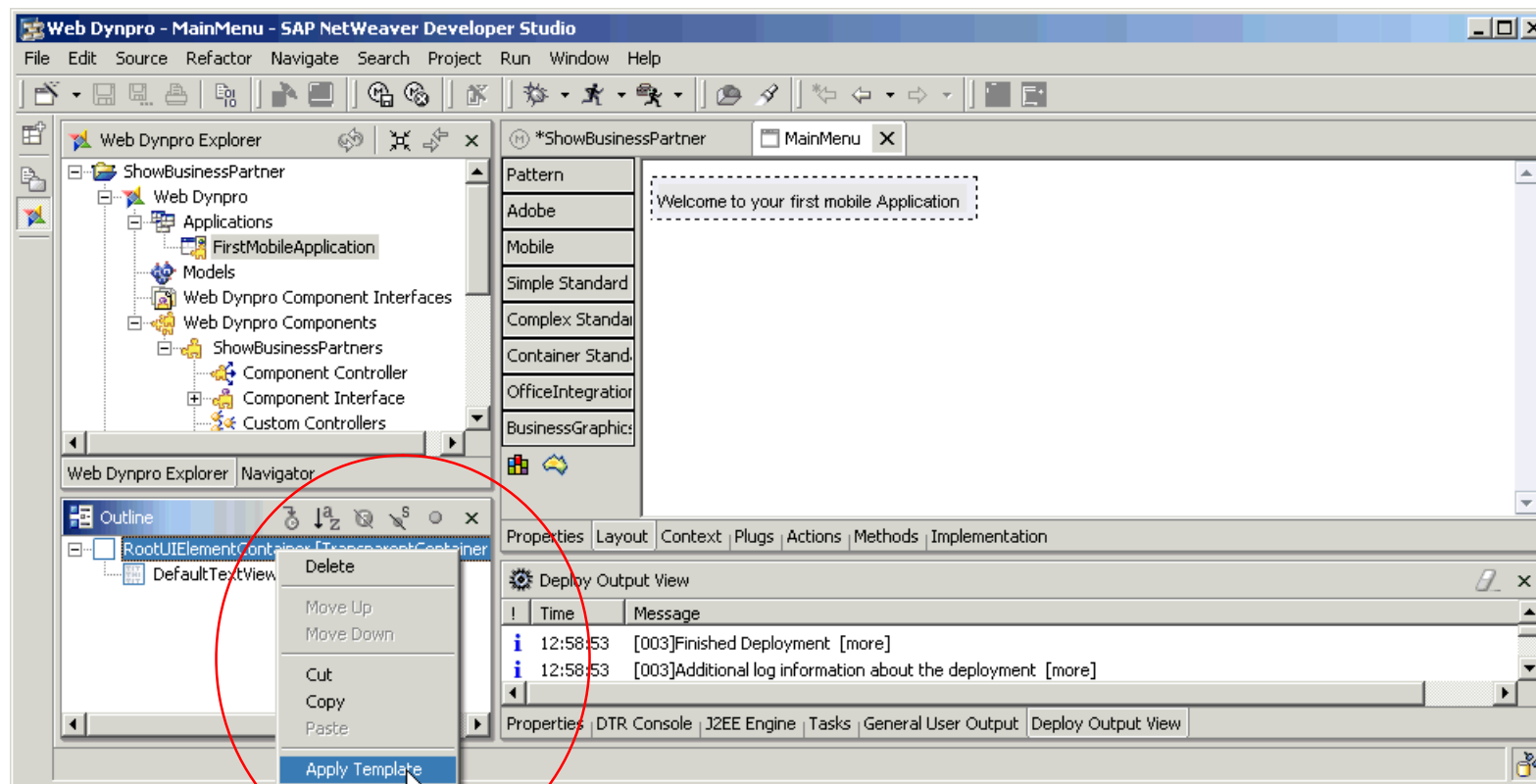


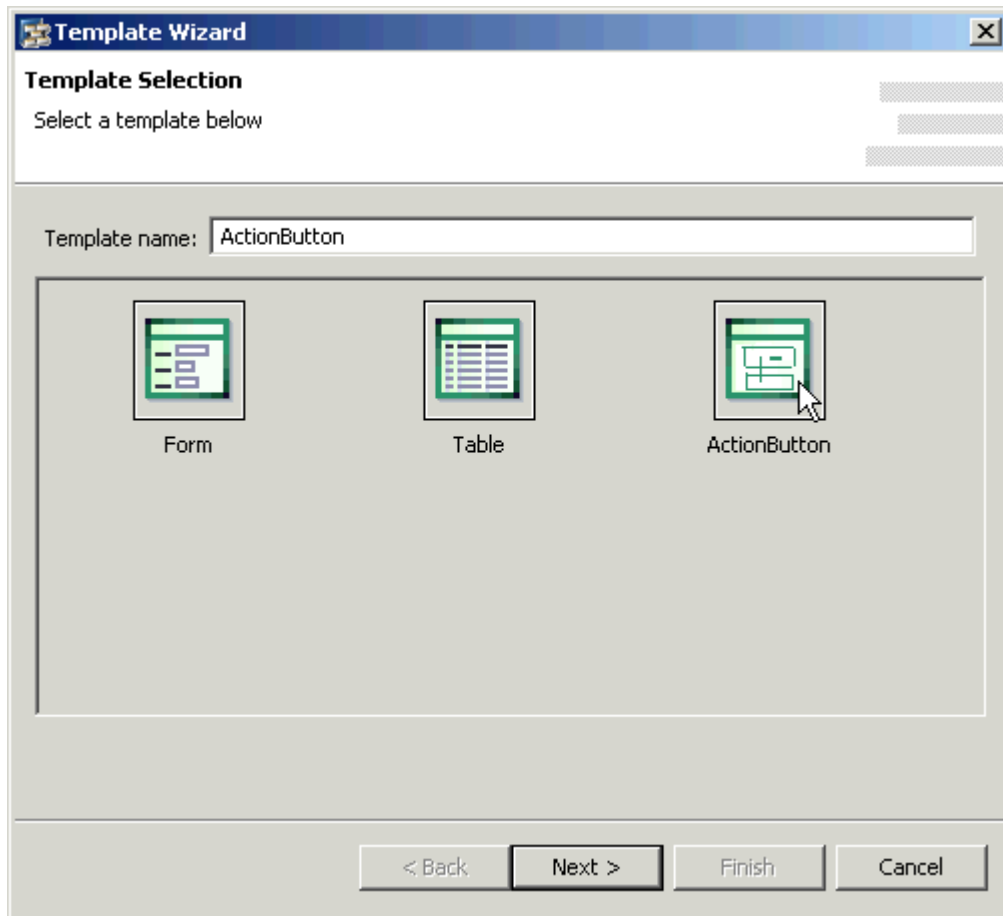
Right mouse click on the outbound plug to show a context menu. Create a navigation link and choose the inbound plug of the new view as the target.

The navigation is shown graphically.



The Views are linked. But we need a button for the user for navigation. A right mouse click gives you the option to apply a template to the view.





Choose an Actionbutton.



Give it a label and fire the Plug for the new view.

Template Wizard

Button Properties
Enter the properties for the button

Button Label: Show Business Data

Action: ShowBusinessData

Event: onActionShowBusinessData

< Back Next >

Template Wizard

Event Handler Properties
Enter the details for the eventhandler associated with the button

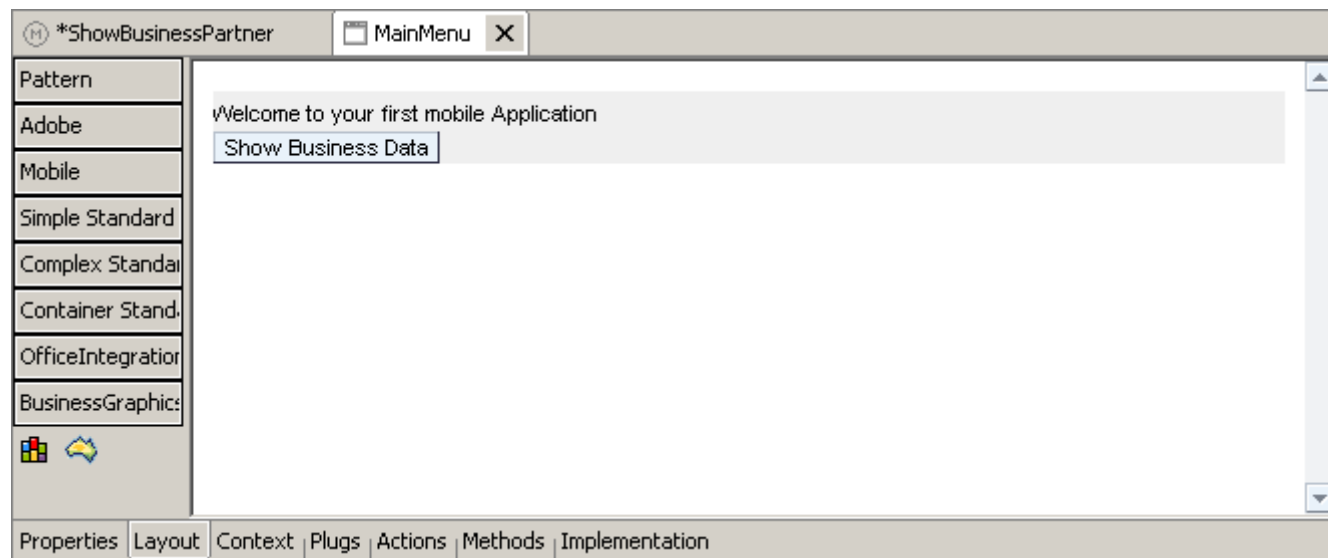
Invoke Methods
 Call Method
Controller: <None>
Methods: <None>

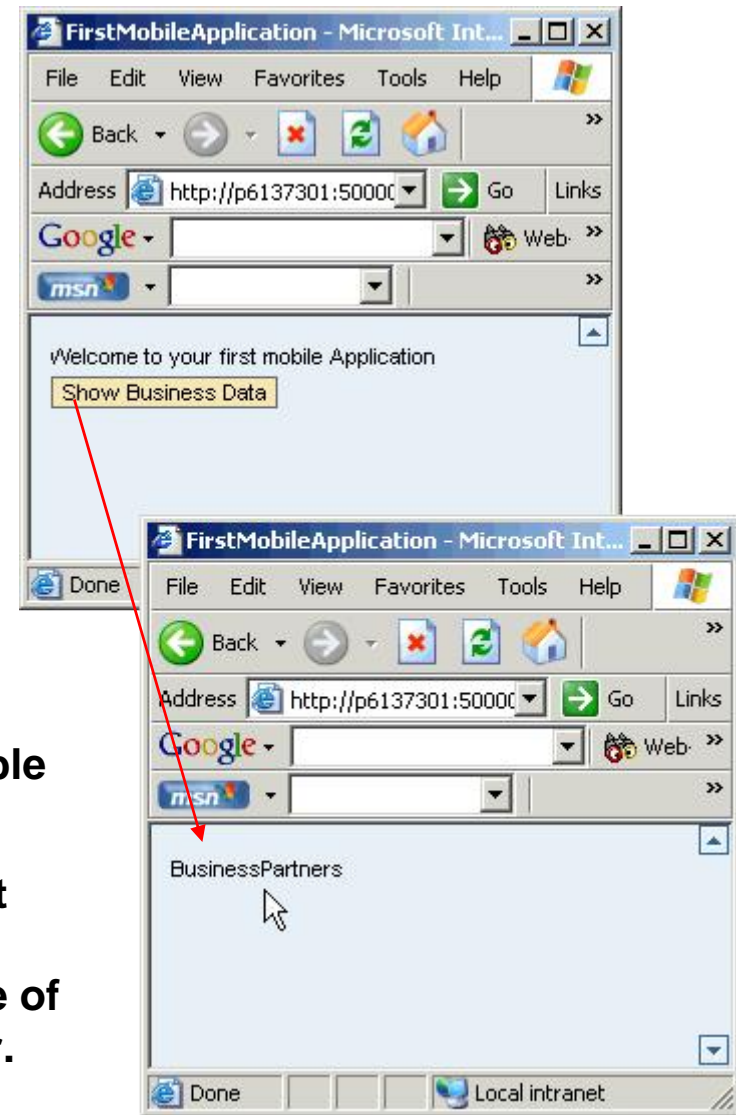
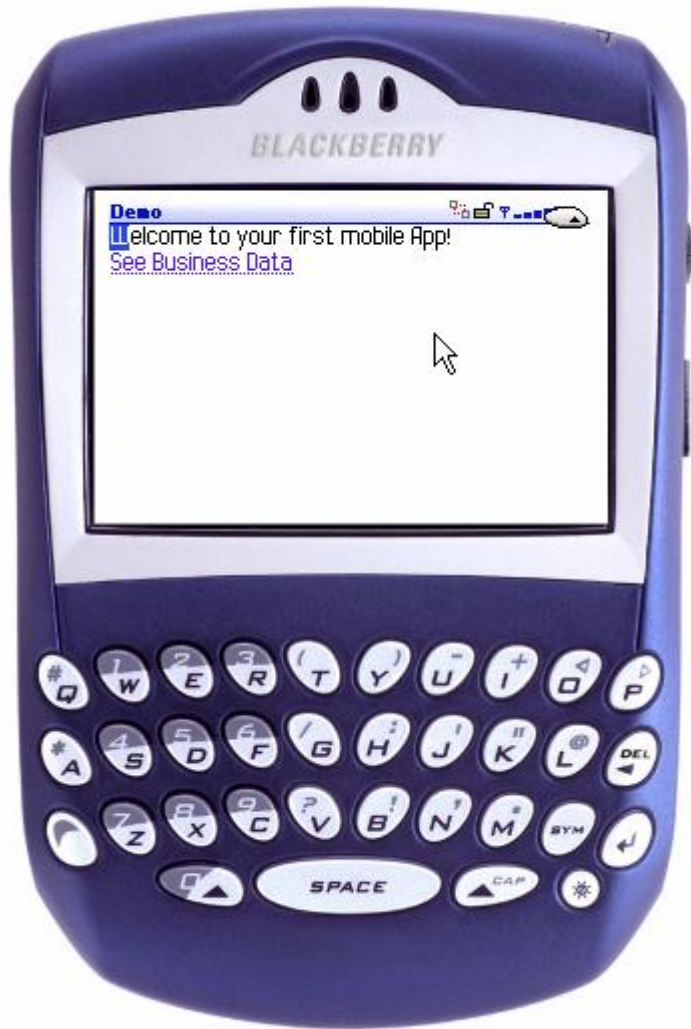
Fire Plug
 Fire Plug
Outbound Plug: doShow New

< Back Next > Finish Cancel



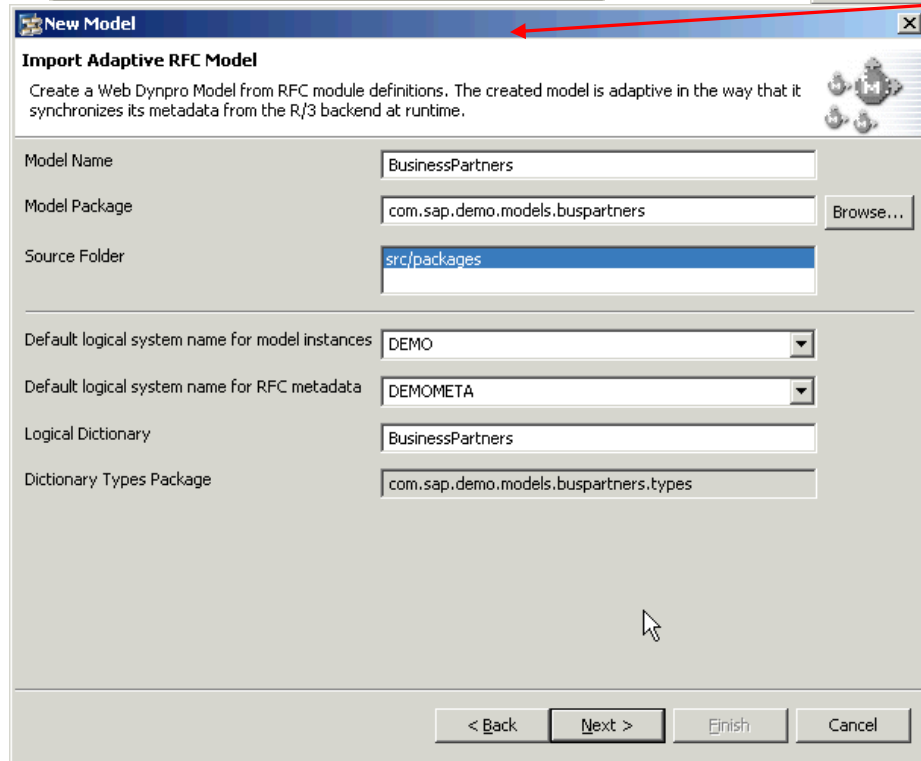
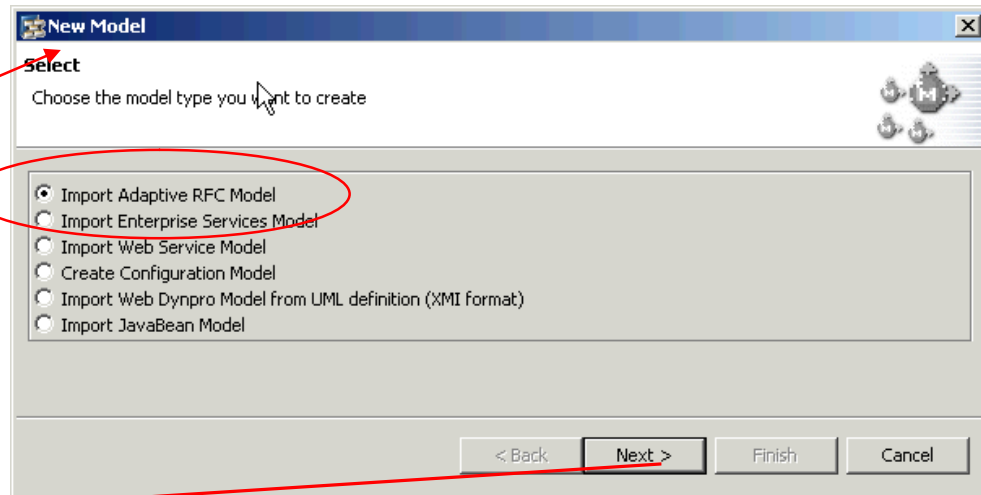
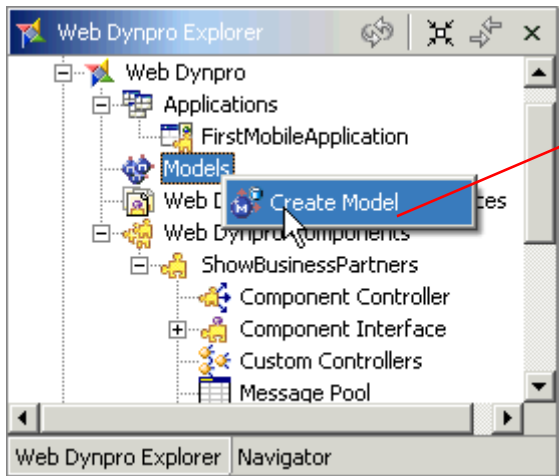
The button is displayed in the view.





The user gets a button and is able to navigate.

Please note that you have not entered one line of java code so far.



Now it is time to create a connection to the backend R/3 system to request data.

Create a new model with a RFC (Remote function call to a SAP system) or choose a connection via Web Services to any other backend system.

New Model

SAP Logon Information
Specify the SAP Server you want to connect to and enter the logon information

Single Server Load Balancing

System: DH6 [SPACE] AFS for Burberry

Message Server: iwd9459.wdf.sap.corp

System Name: DH6

Group: SPACE

User Account

Client: 781

Logon Name: ahland

Password: *****

Language: EN

< Back Next >

In the next step, enter the connection information, and connect to the backend system.

New Model

SAP Logon Information
Specify the SAP Server you want to connect to and enter the logon information

Single Server Load Balancing

System: DH6 [SPACE] AFS for Burberry

Message Server: iwd9459.wdf.sap.corp

System Name: DH6

Group: SPACE

User Account

Client: 781

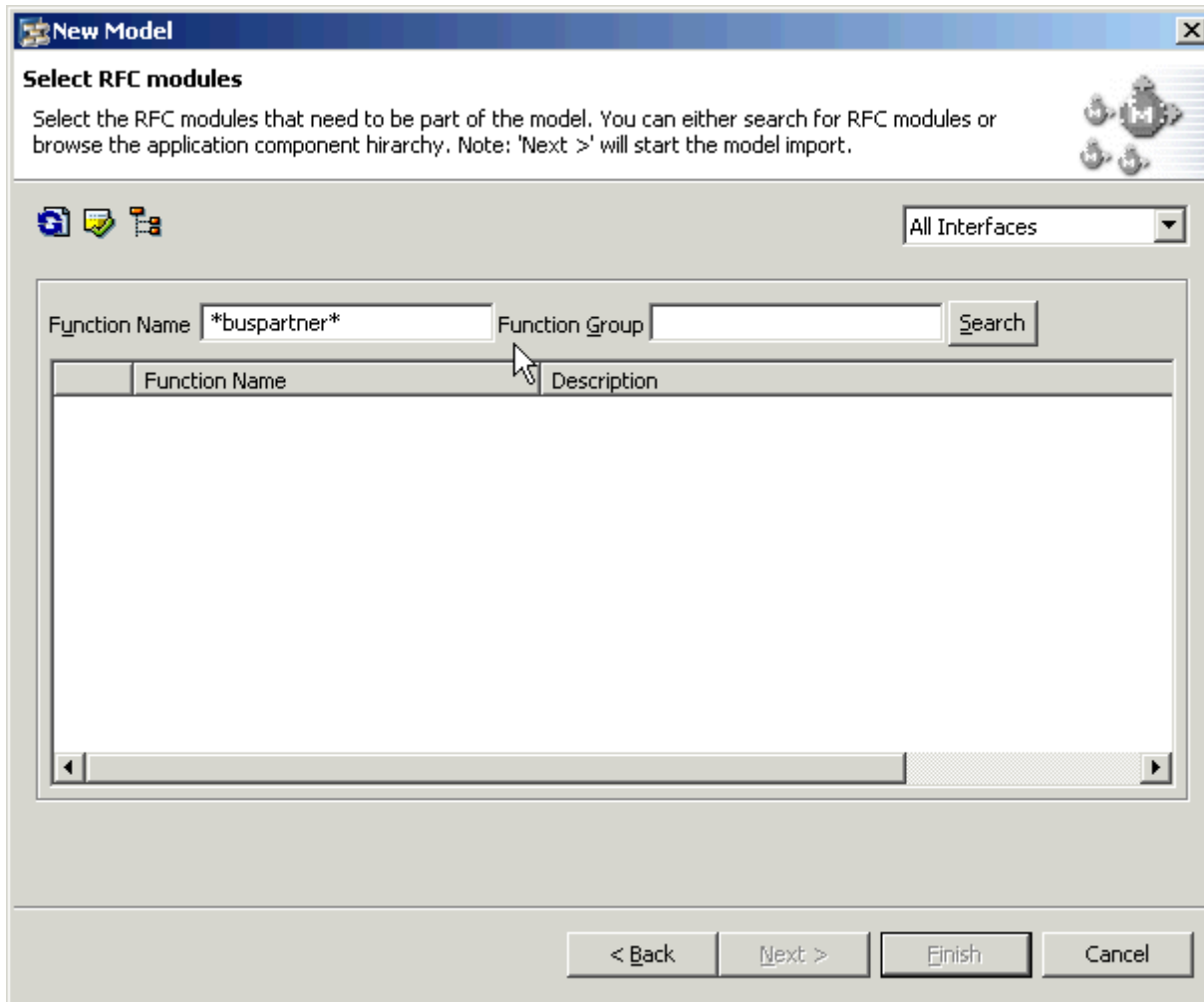
Logon Name: ahland

Password: *****

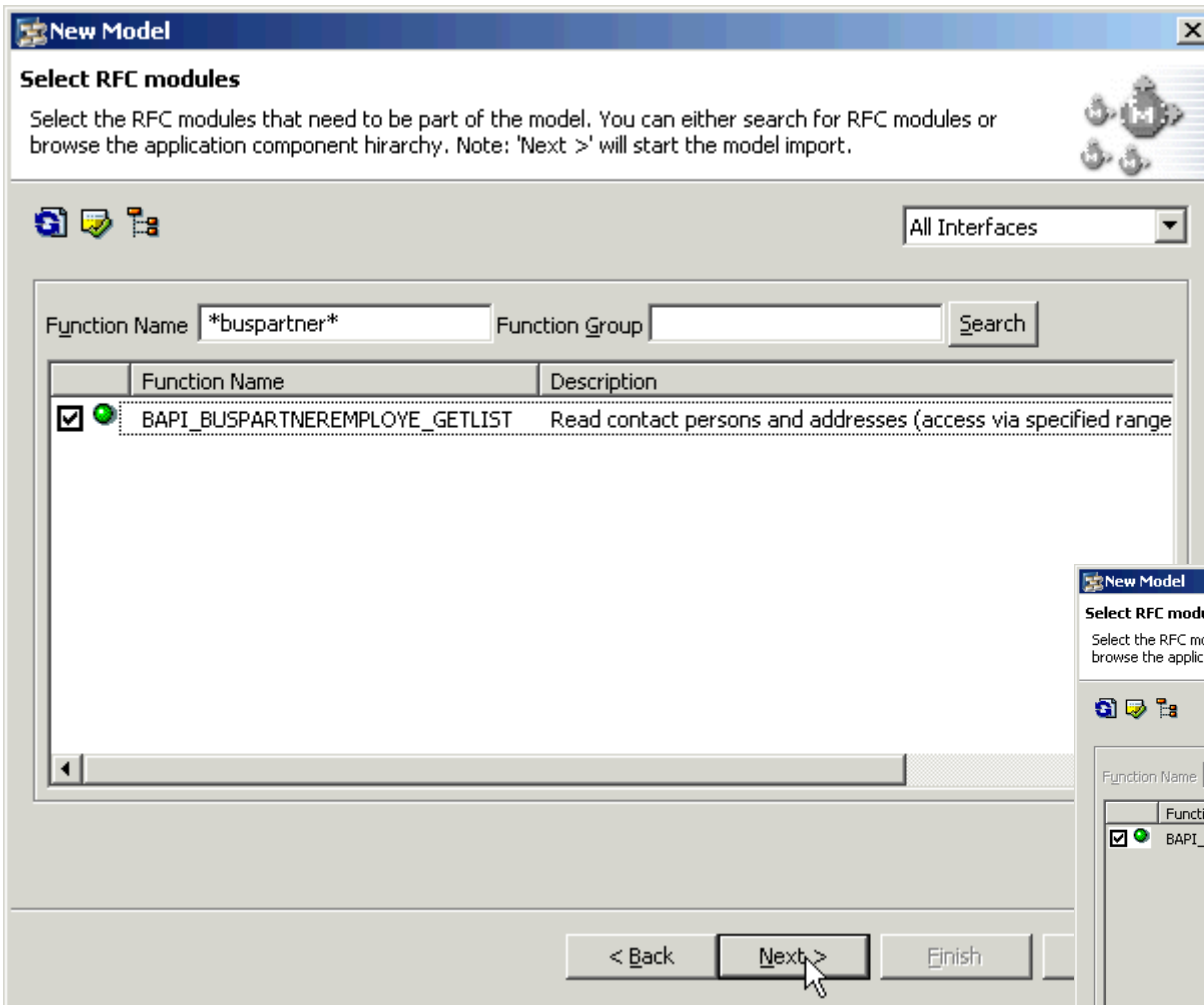
Language: EN

Connecting to R/3 System

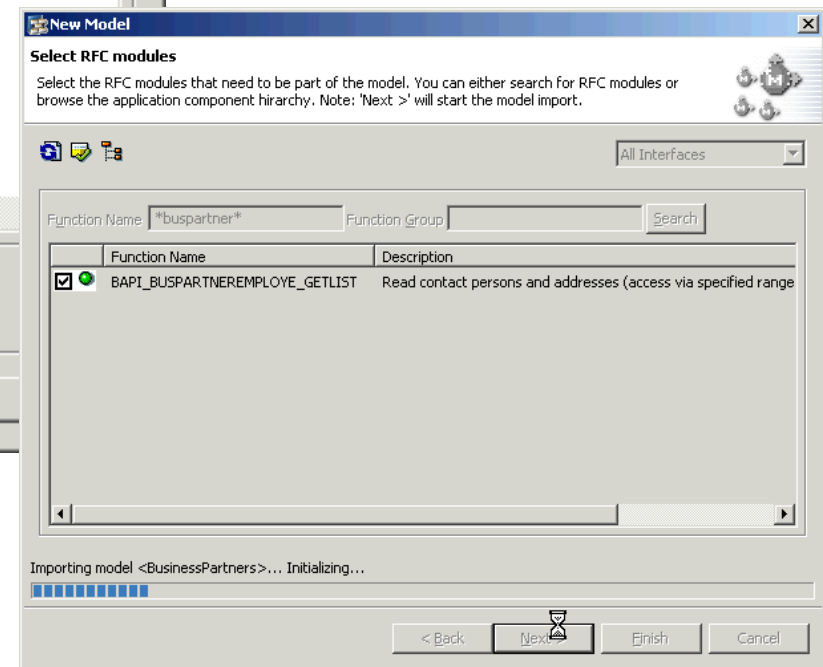
< Back [Progress Bar] Finish Cancel

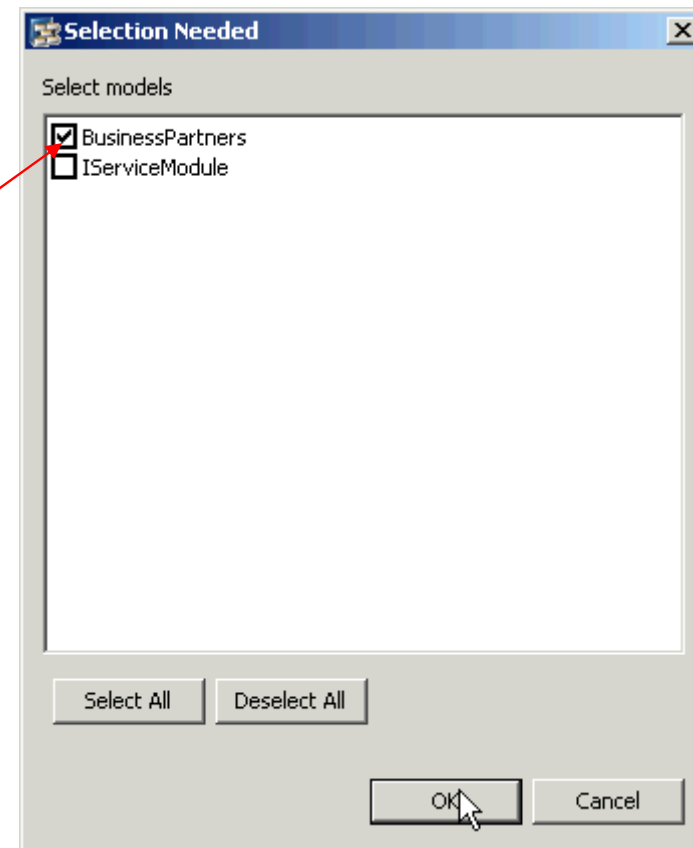
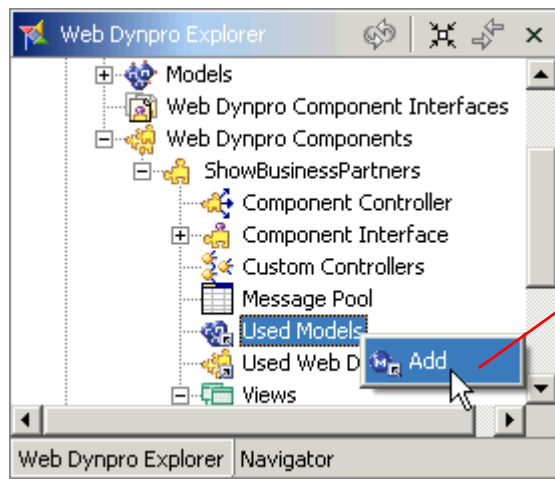


Search for the remote function module you want to call ...



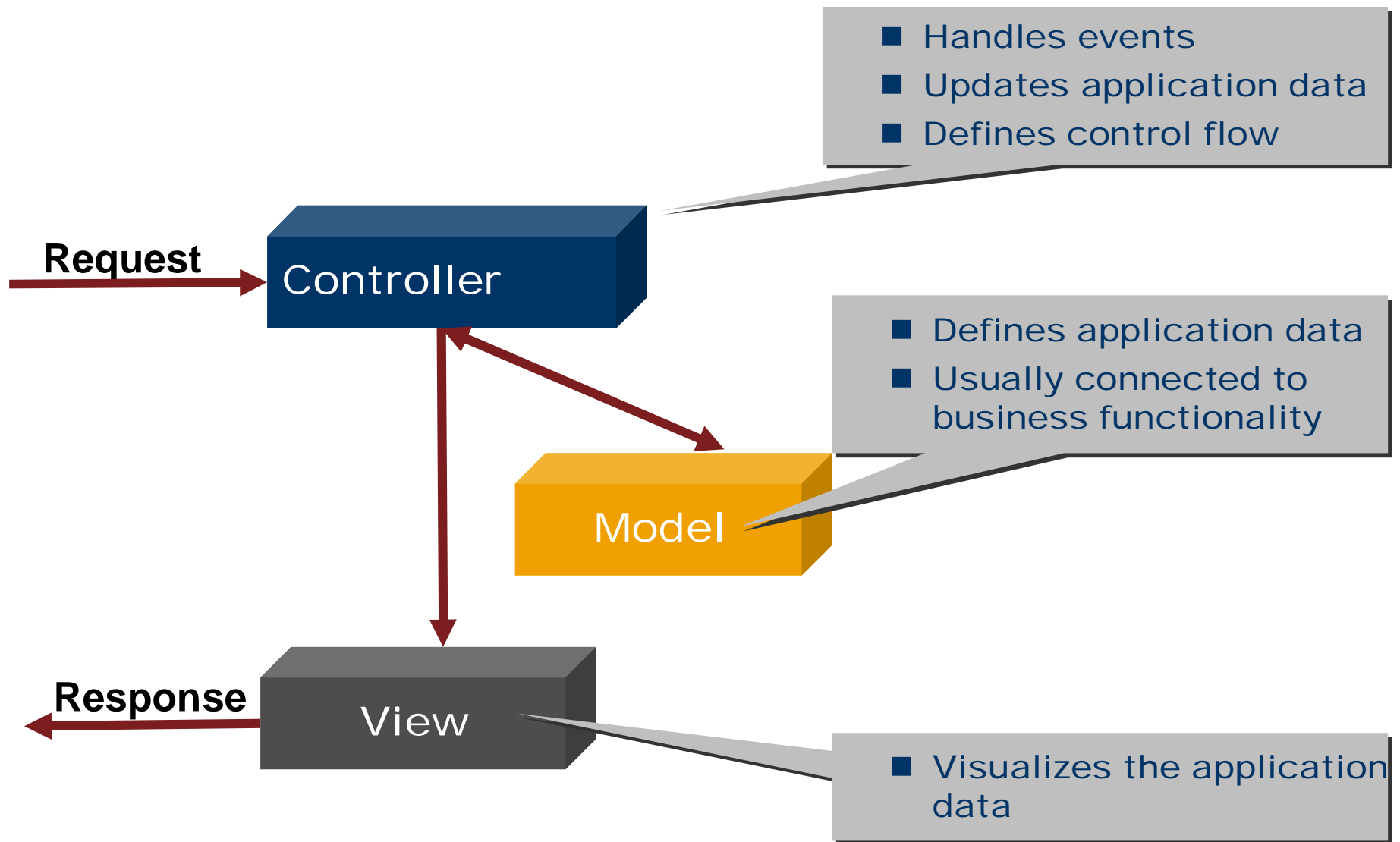
and select it. All the needed java classes and dictionary entries are created by Web Dynpro, and this give you a frameset to work with the data.



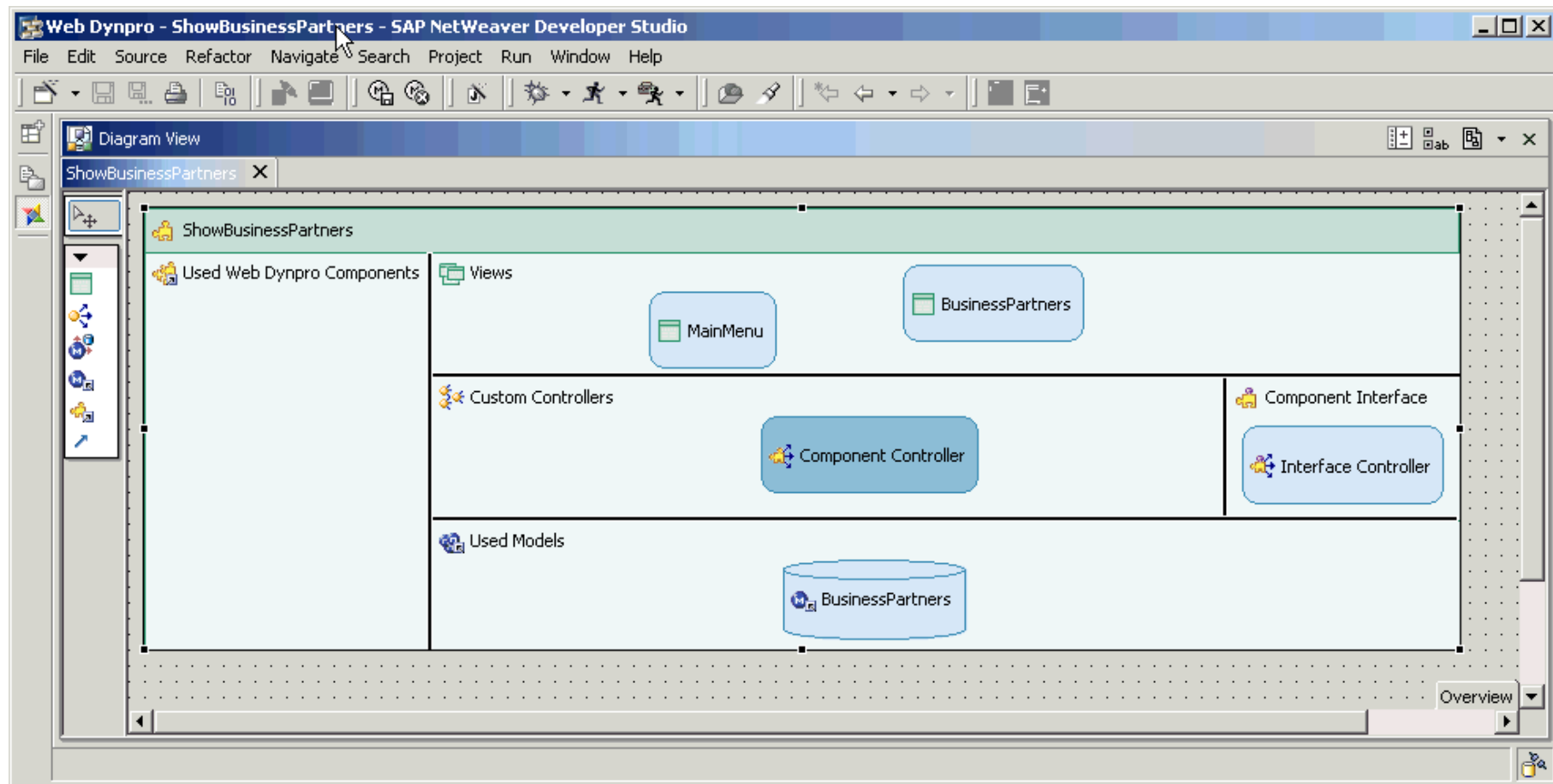


Use the new model in the component.

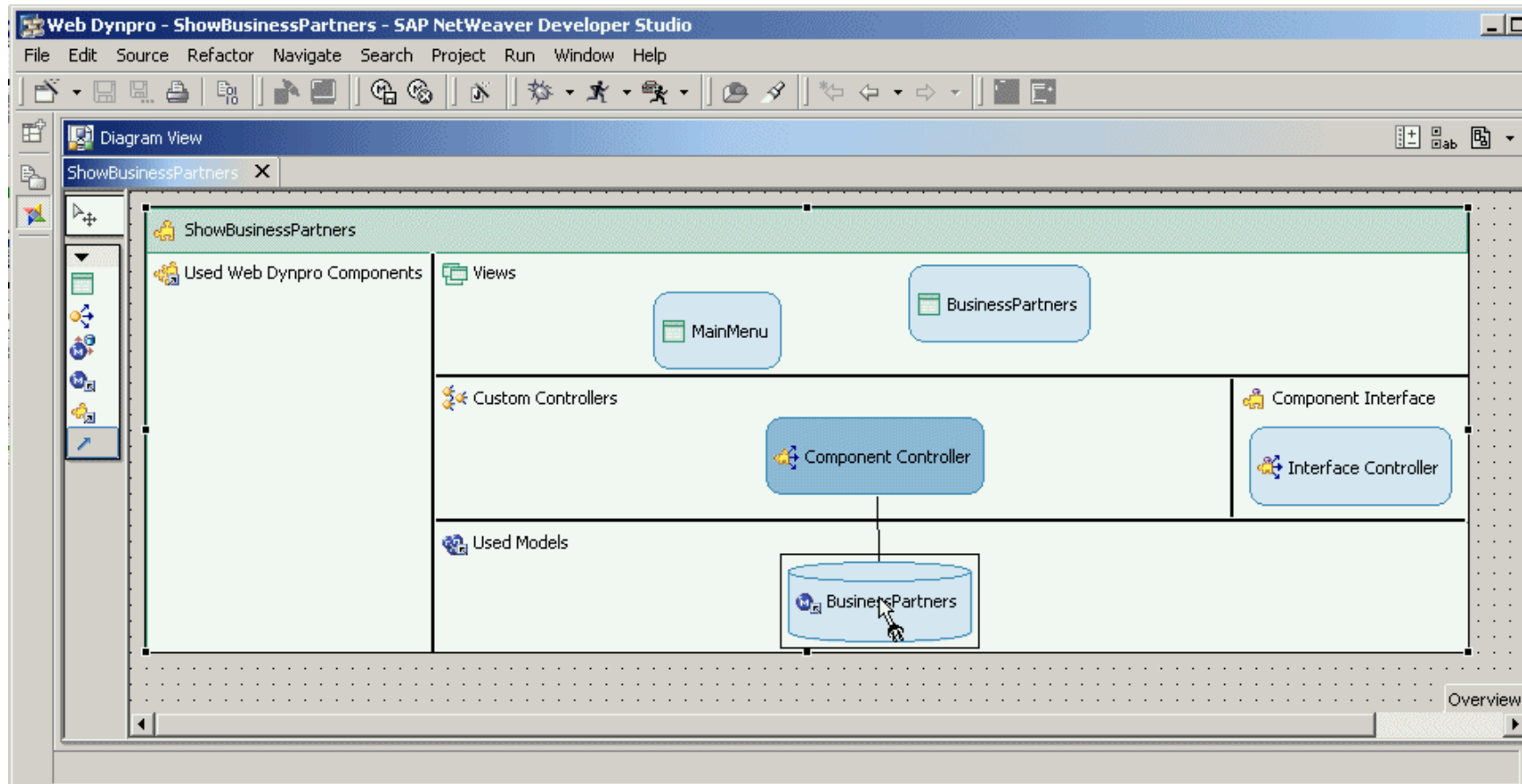
As a reminder Model View Controller (MVC)



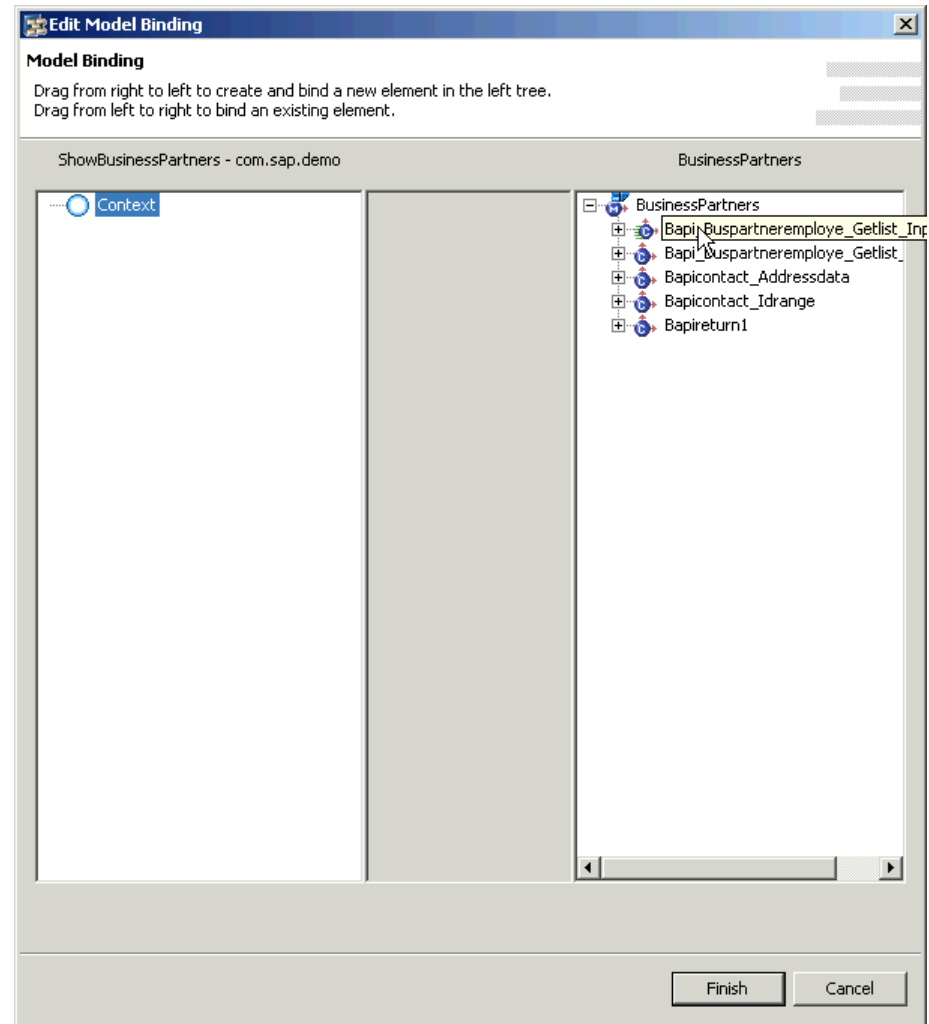
Double clicking on the component you will see this MVC-concept for your application



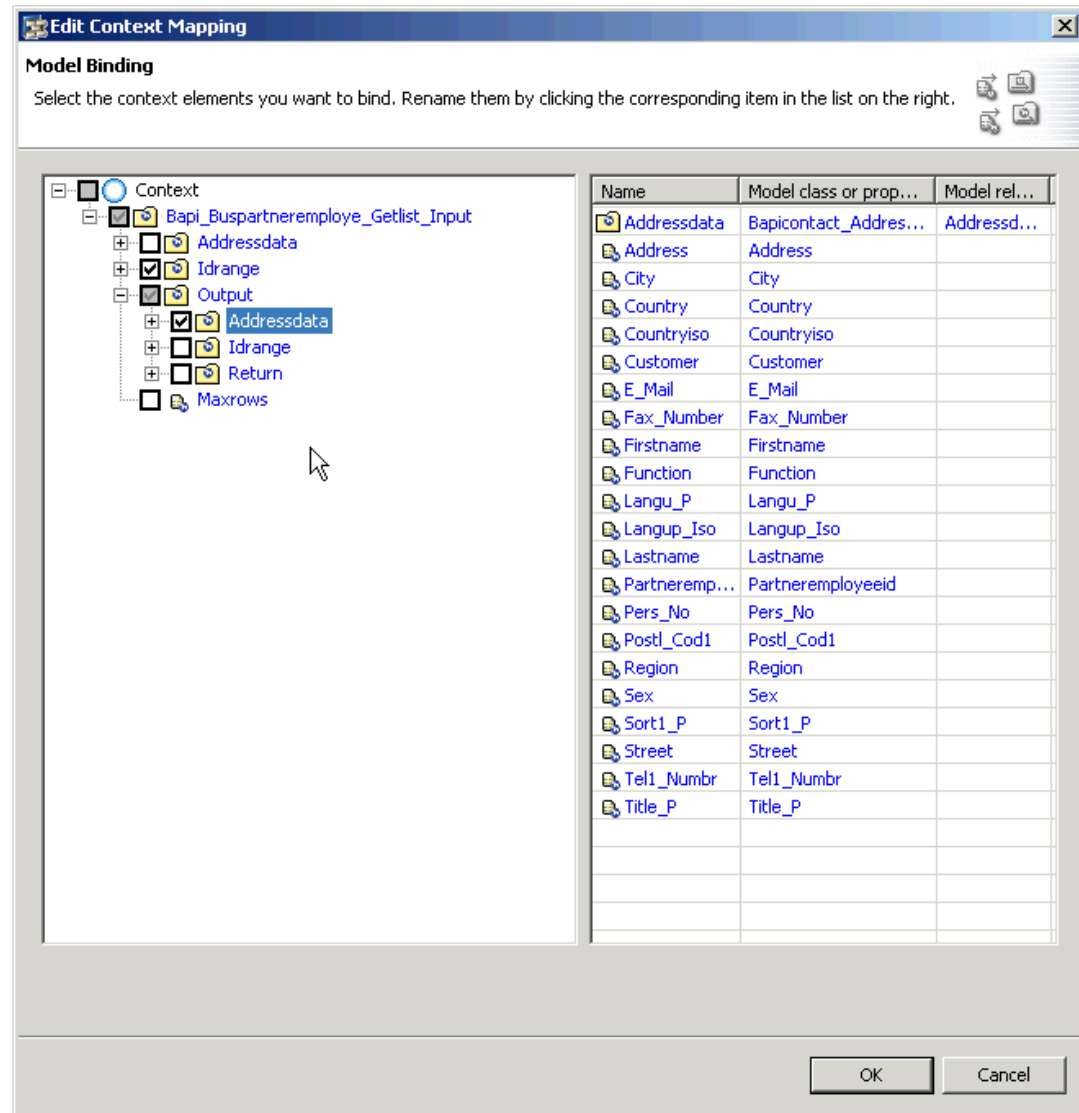
Drag and drop a connection between the Component Controller and the model.



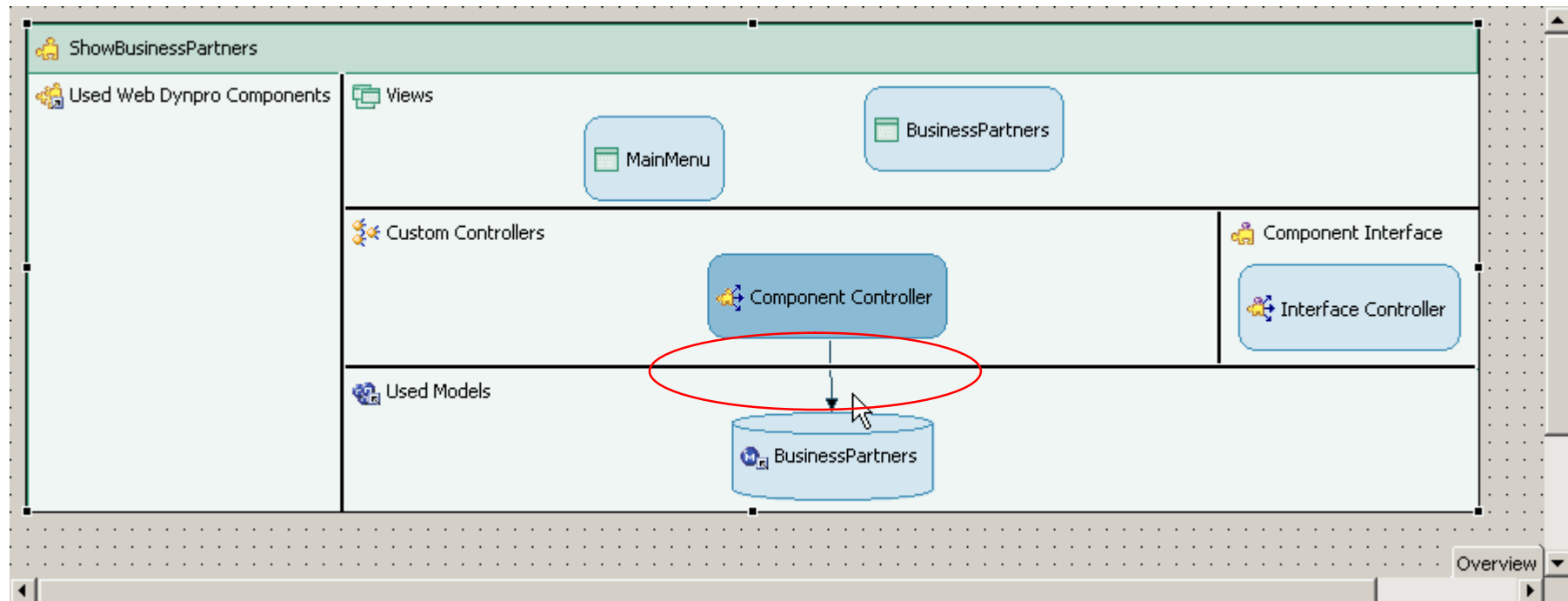
This opens a wizard showing parts of the rfc which can be used.

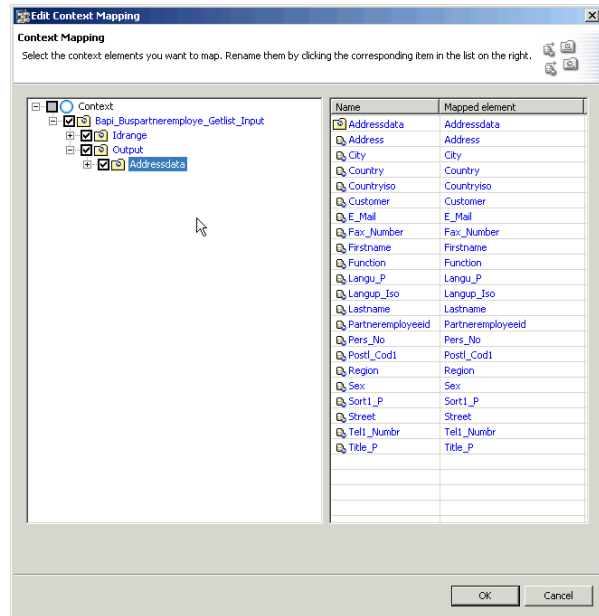


including the fields.

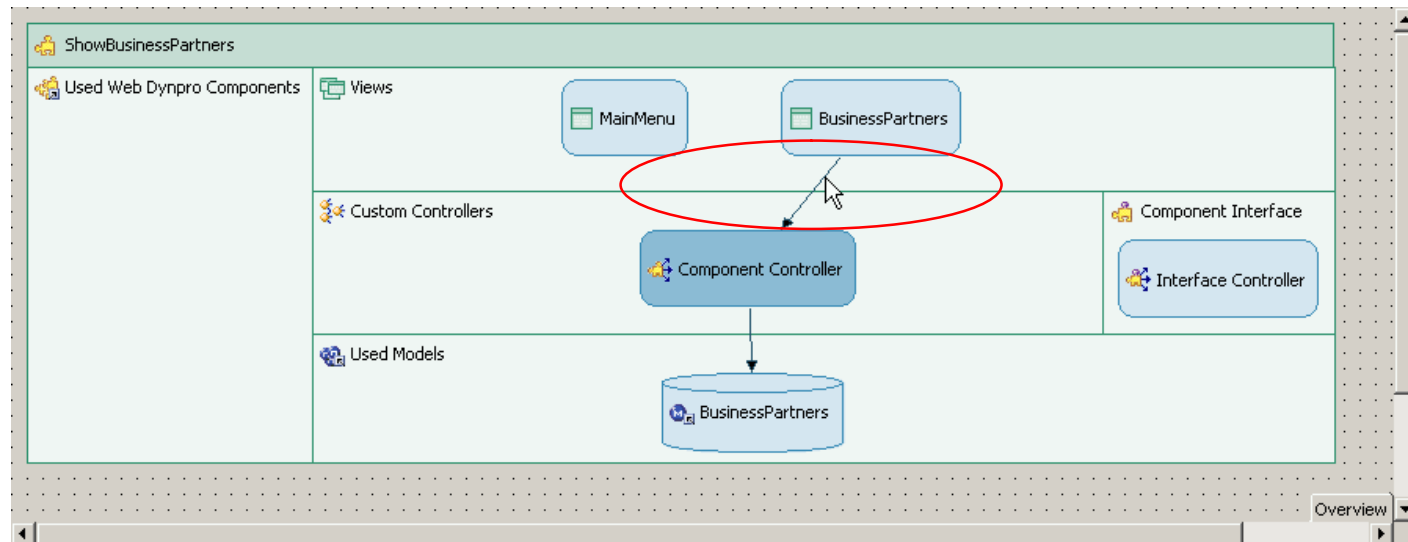


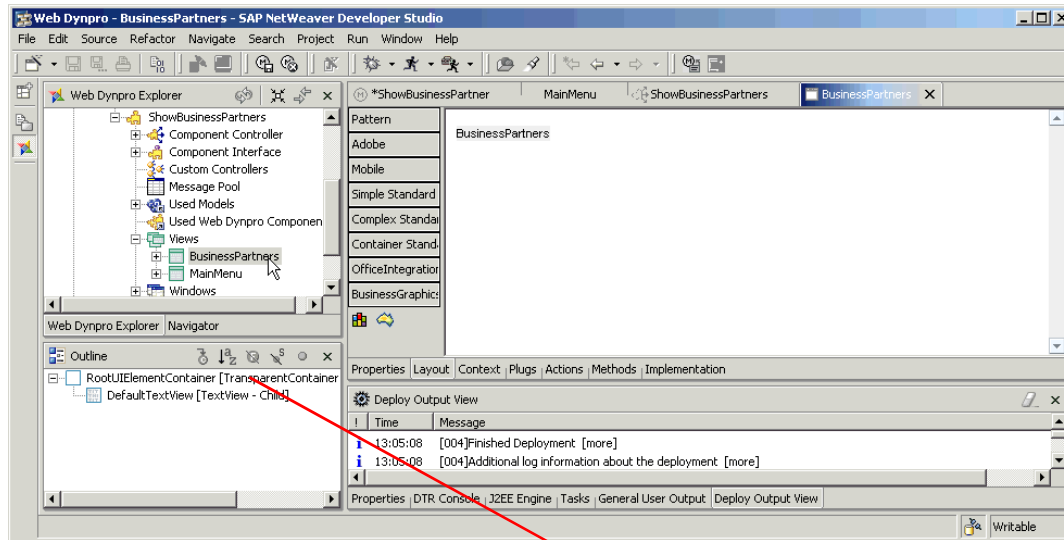
On completing the wizard, a connection between the component controller and the model is shown graphically



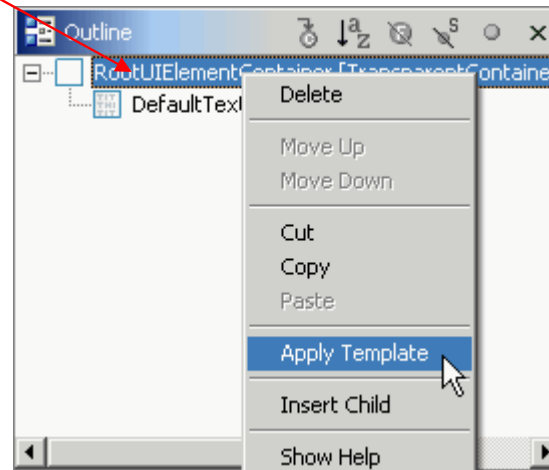


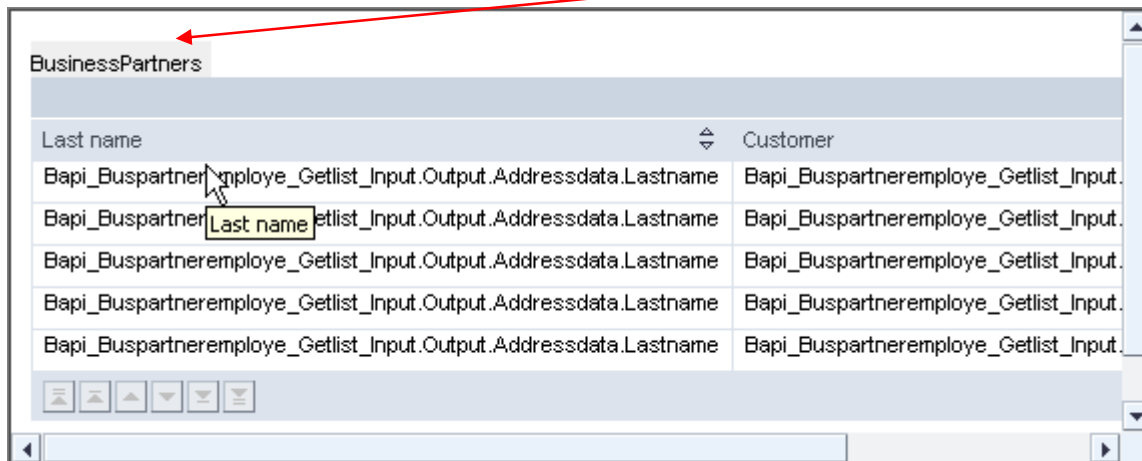
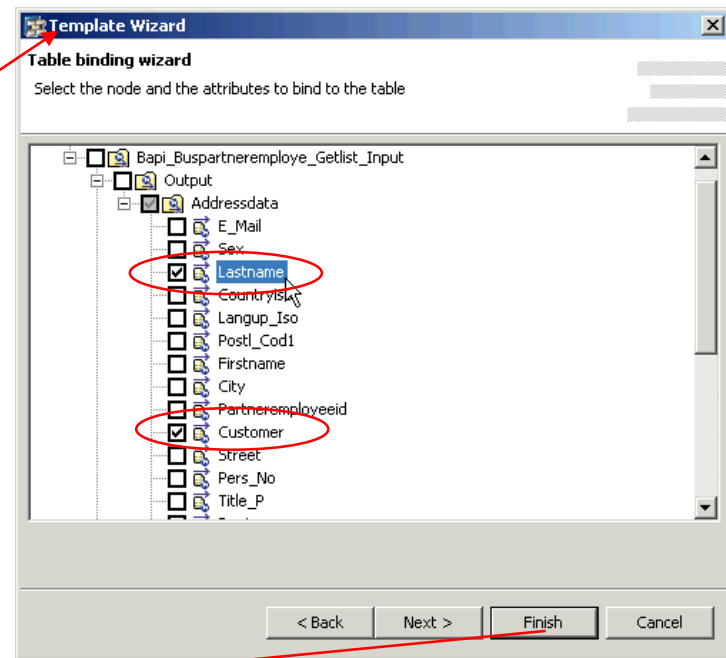
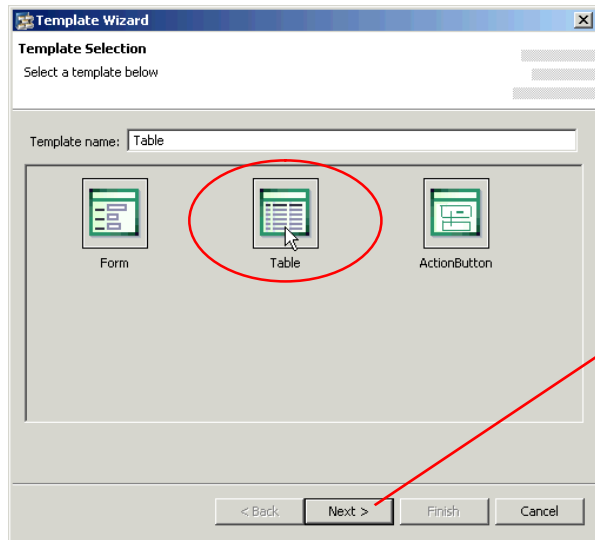
Proceed in the same manner to connect the view, which will show the data, and the Component Controller.





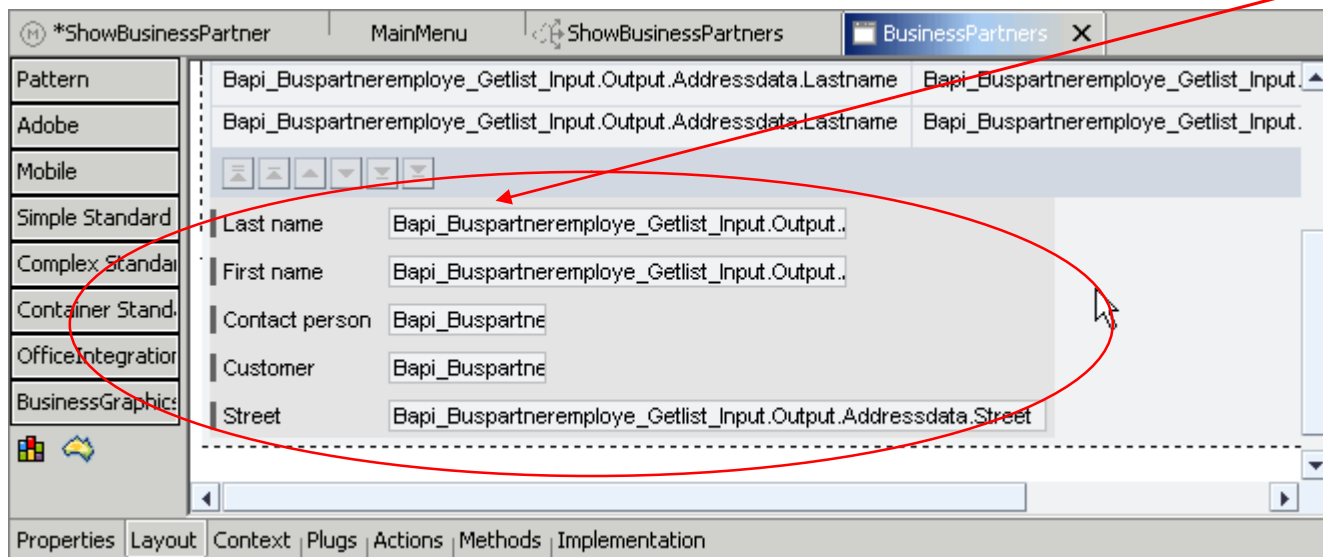
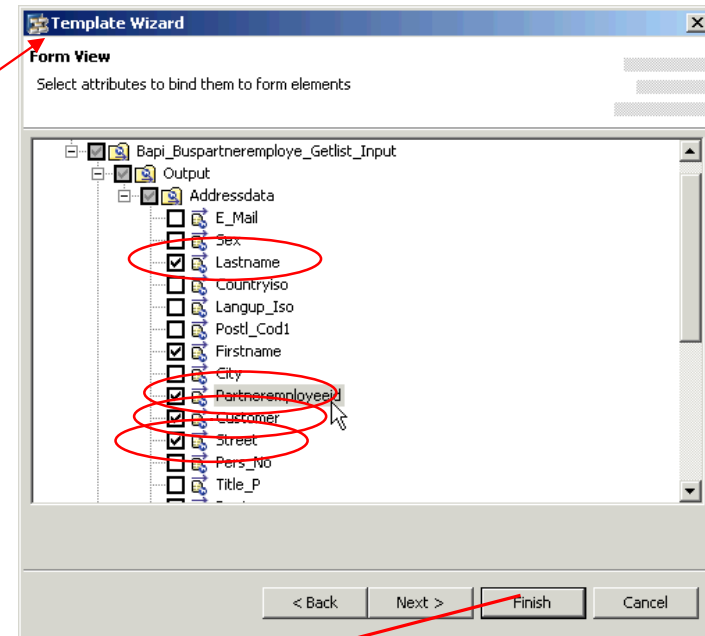
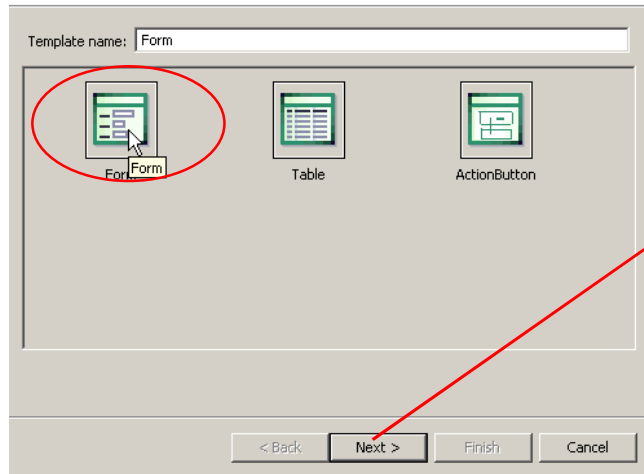
Jump to the view which will show the data and apply a template to display the data as a table.





In the wizard, mark the fields to be displayed. Finishing the wizard creates a table in the view.

Template Selection
Select a template below



Proceed in a similar manner to add some fields for a detail view of the selected business partner in the table.

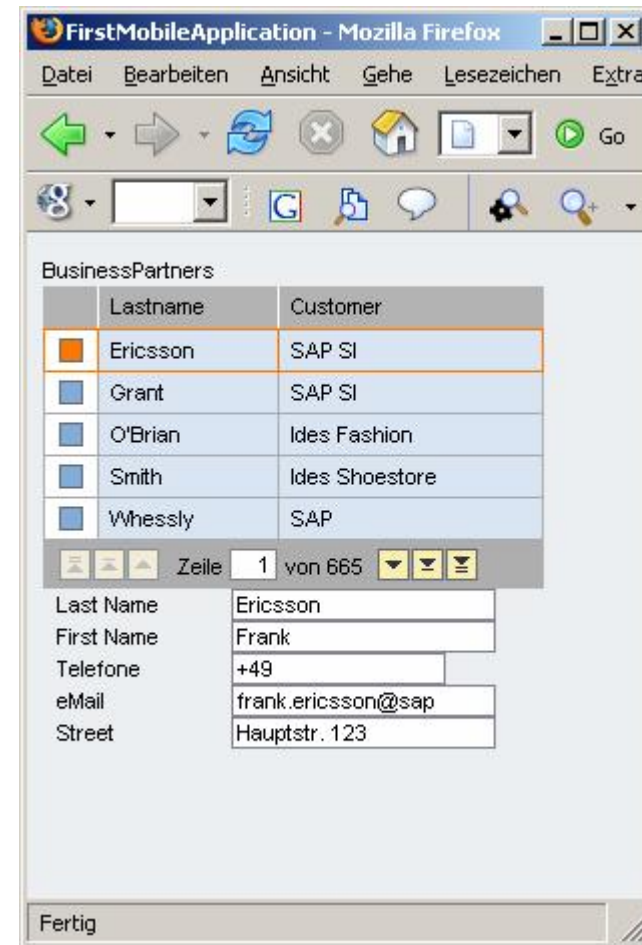
Now a little bit of java code is required to initialize the function call, to bind the data and to catch exceptions. Place the code shown below in the wdInit method of the controller.

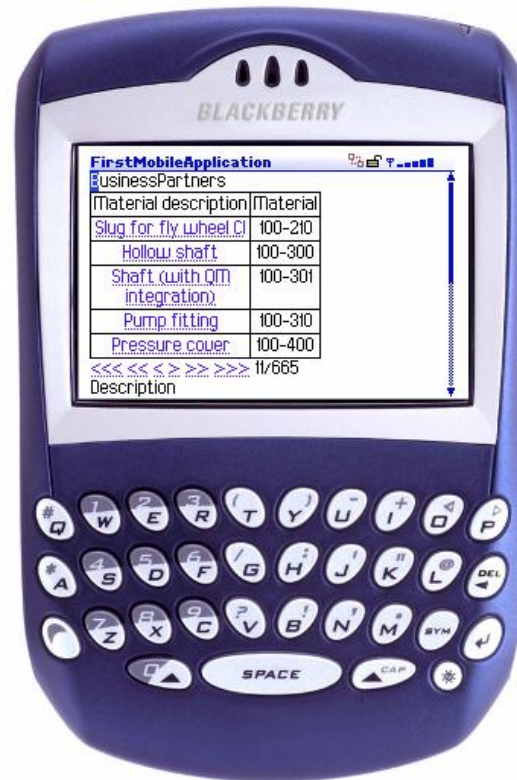
```
Alm_Merep_070_Getlist_Input materialInput = new Alm_Merep_070_Getlist_Input();
wdContext.nodeAlm_Merep_070_Getlist_Input().bind(materialInput);

try {
    materialInput.setUser("AHLAND");
    materialInput.execute();
    wdContext.nodeOutput().invalidate();
}
catch (Exception e) {
    wdComponentAPI.getMessageManager().
    raiseException("BAPI Call failed "+e.toString(),false);
}
```

Deploying the application shows you a list of business partners. Clicking on a row automatically changes the detail view in the lower part of the page.

What is going on? Clicking on a new row fires a round trip to the server and changes the current selection. The detail view shows the current selection which is updated automatically.





On a Blackberry it could look like this.