

# Models Solutions



**Chapter:** Models. EJB

**Theme:** Using Web Dynpro to access EJBs

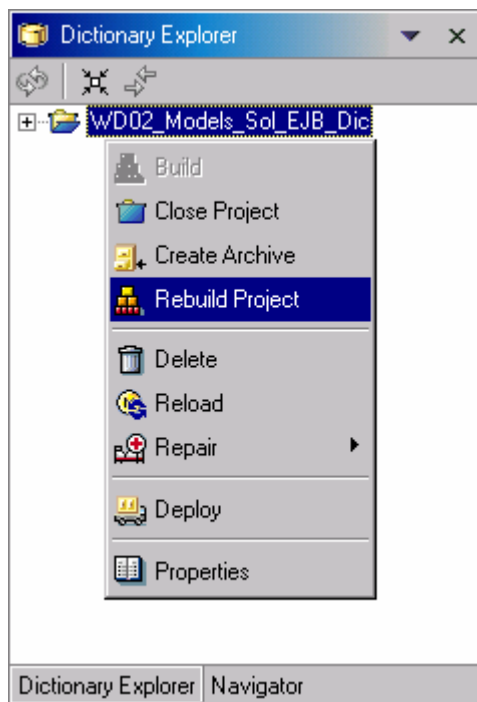


At the end of this Exercise, you are able to:

- Access EJB functionality from Web Dynpro

## 4 Developing, Step-by-Step

### 4-1 Deploy the predefined Dictionary archive.



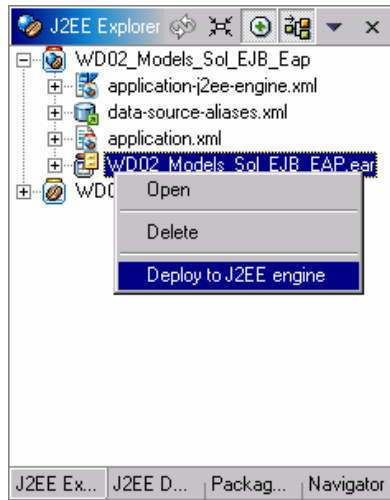
Open the *Dictionary* perspective.

Open the project  
*WD02\_Models\_Sol\_EJB\_Dic*.

From the context, choose  
*Rebuild Project* and *Deploy*.

This will deploy the predefined metadata of the database table *TMP\_SOL\_CARRENTAL* to the corresponding database. The table will then be created on the database.

#### 4-2 Deploy the predefined EJB components.



Open the J2EE development perspective.

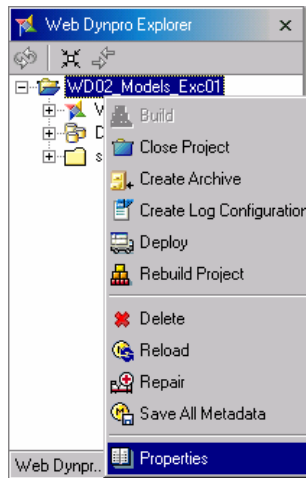
From the package explorer, open the projects *WD02\_Models\_Sol\_EJB\_Eap* and *WD02\_Models\_Sol\_EJB\_Mod*.

From the context of *WD02\_Models\_Sol\_EJB\_Eap / WD02\_Models\_Sol\_EJB\_Eap.ear* choose *Deploy to J2EE engine*

This will deploy the predefined EJBs to the SAP Web Application server.

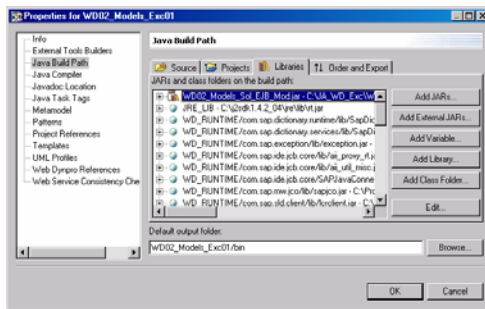
#### 4-3 Customize the project settings.

##### 4-3-1 Add the EJB-jar file to the WD project



In the Web Dynpro Explorer, open the context of the project node *WD02\_Models\_Exc\_EJB*.

Choose *Properties*.

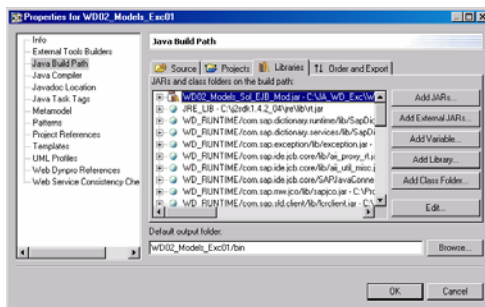


In the wizard that appears, choose the entry *Java Build Path* and the tab *Libraries*.

Make sure, that the file *WD02\_Models\_Sol\_EJB\_Mod.jar* has been added.

If not press the button “Add external jars” and select the jar file. It should be available at *<Coursepath> \ WD02\_Models\_Sol\_EJB\_Mod.jar*. Choose the *Order and Export* tab. Move the EJB jar file to the top.

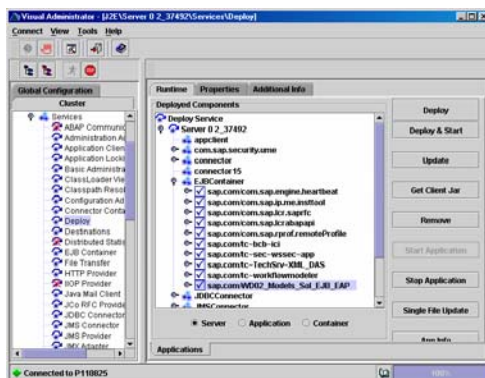
#### 4-3-2 Define the Sharing Reference of your Web Dynpro Project



In the *Properties* wizard, choose the entry *Web Dynpro References* and then the tab *Sharing references*.

Add a new Sharing reference with the following syntax:  
*<vendor name> / <name of the ear file without extension>*

In our case this would be  
*sap.com/WD02\_Models\_Sol\_EJB\_EAP*



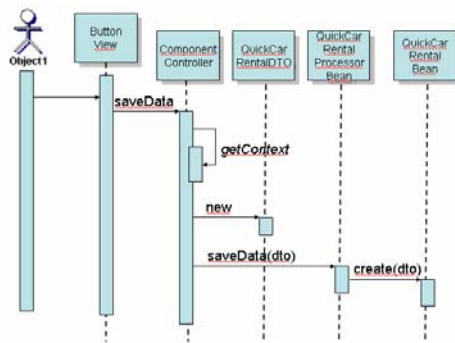
Note: This is the name under which the EJB application is stored on the Web Application server.

You find an entry in the Visual Administrator tool under *Server 0 / Services / Deploy* tab *Runtime* entry *Server 0 / EJBContainer* as shown in the graphic on the left side.

Save the new project data by choosing *Save All Metadata* from the toolbar.

#### 4-4 Complete the SAVE action.

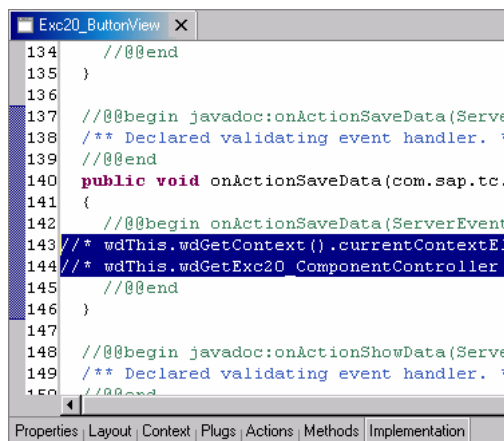
##### Overview



Please take a look at the exercise to get an overview over the process flow of the save action.

The necessary code is completely predefined. You only have to uncomment the corresponding lines in the source code.

##### 4-4-1 Exc\_EJB\_ButtonView



Open the Implementation tab of *Exc\_EJB\_ButtonView*.

Navigate to method

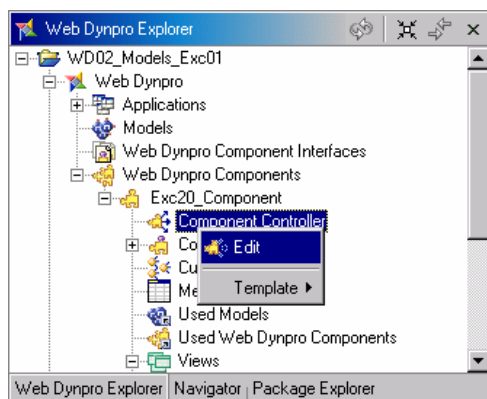
*onActionSaveData(...)*.

This method is called, when a user chooses the *Save* button.

Remove “*/\*\**” from the source code lines.

Save the new project data by choosing *Save All Metadata* from the toolbar.

##### 4-4-2 Exc\_EJB\_Component Controller



Open the tab *Implementation* of *Exc\_EJB\_Component*.

Remove “*/\*\**” from the import statements:

```
import
    com.sap.training.wd.sol.ejb.QuickCarRentalProcessorLocal;
import
    com.sap.training.wd.sol.ejb.QuickCarRentalProcessorLocalHome;
import com.sap.training.wd.sol.ejb.util.QuickCarRentalDTO;
import com.sap.training.wd.sol.ejb.util.QuickCarRentalException;
```

Navigate to the end of the class definition and remove “//\*” from the attribute definition:

```
//@@begin others
private QuickCarRentalProcessorLocal reservationBean;
//@@end
```

Navigate to *saveData(...)* and remove all “//\*” at the beginning of the source code lines within the method.

Navigate to *initializeReservationBean(...)* and remove all “//\*” at the beginning of the source code lines within the method.

As you can see, the method checks, if an instance of the Stateless Session Bean exists (if not, a new instance is created). Then the method gets the content of the input fields from the Component Controllers context. After that, a DTO is created and filled with the input values. Then the *saveData(dto)* method of the Stateless Session Bean is called. The Session Beans passes the DTO to the CMP Entity Bean. The CMP Entity Bean then causes the Web Application Server to create a corresponding entry on the database table.

Save the new project data by choosing *Save All Metadata* from the toolbar

## 5 Building, Deploying, and Running, Step-by-Step

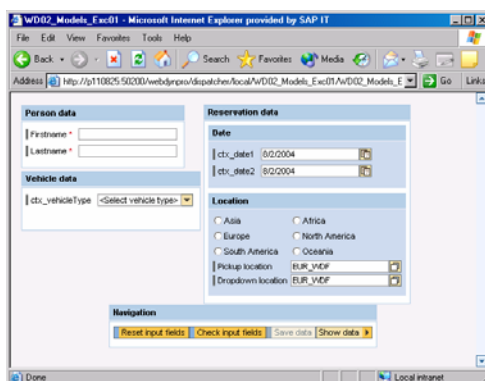
### 5-1 Deploy and run the Web Dynpro application.

In the Web Dynpro Explorer:

Expand the nodes *WD02\_Models\_Exc\_EJB / Web Dynpro / Applications*.

Open the context menu for *WD02\_Models\_EJB*.

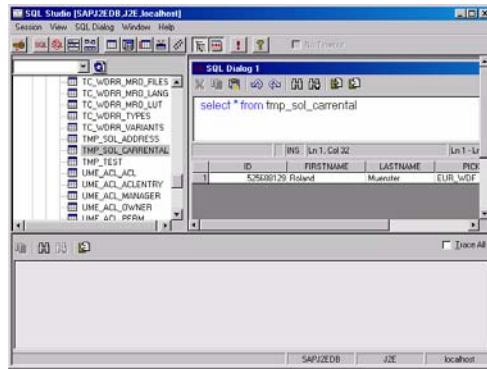
To deploy and run the application, choose *Deploy new Archive and Run*



In the application that appears, type in some values, check the input fields and press the *Save* button.

The content of the input fields should be saved to the database.

Open SQLStudio to control if the application works successfully.



Open the SQL Studio and insert the following SQL statement to see all data records:

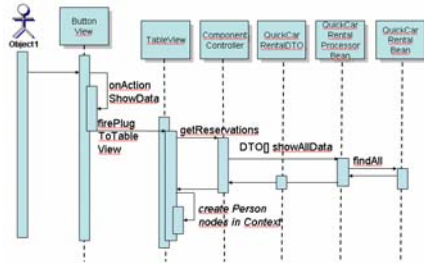
```
select * from  
tmp_sol_carrental
```

Press F8 to start the selection.

## 6 Optional: Complete the SHOW and DELETE actions

### 6-1 Complete the SHOW action.

#### Overview



Please take a look at the exercise to get an overview over the process flow of the action SHOW.

The necessary code is completely predefined. You only have to uncomment the corresponding lines in the source code.

#### 6-1-1 Exc\_EJB\_TableView

Open the Implementation tab of *Exc\_EJB\_TableView*.

Navigate to method *onPlugFromNavigationView()*.

Remove “`/**`” from source code lines.

Save the new project data by choosing *Save All Metadata* from the toolbar.

#### 6-1-2 Exc\_EJB\_Component Controller

Open the Implementation tab of *Exc\_EJB\_Component Controller*.

Navigate to method *getReservations(...)*.

Remove “`/**`” from source code lines.

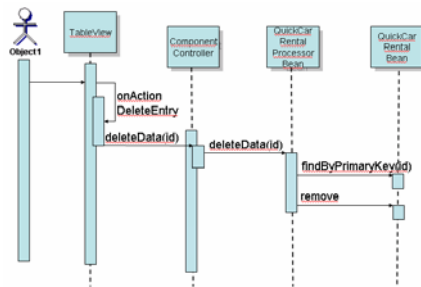
Save the new project data by choosing *Save All Metadata* from the toolbar.

### 6-2 Deploy and run the application.

Take a look at exercise 5-1.

### 6-3 Complete the DELETE action.

#### Overview



Please take a look at the exercise to get an overview over the process flow of the delete action.

The necessary code is completely predefined. You only have to uncomment the corresponding lines in the source code.

#### 6-3-1 Exc\_EJB\_TableView

Open the Implementation tab of *Exc\_EJB\_TableView*.

Navigate to method *onActionDeleteEntry()*.

Remove “*/\*\**” from source code lines.

Save the new project data by choosing *Save All Metadata* from the toolbar.

#### 6-3-2 Exc\_EJB\_Component Controller

Open the Implementation tab of *Exc\_EJB\_Component Controller*.

Navigate to method *deleteData(...)*.

Remove “*/\*\**” from source code lines.

Save the new project data by choosing *Save All Metadata* from the toolbar.

### 6-4 Deploy and run the application.

Take a look at exercise 5-1.