

OPEN UP POSSIBILITIES WITH MOBILE WEB DYNPRO ONLINE

UP151

Exercise / Solution

KARIN SCHATTKA, SAP AG
CHRIS WHEALY, SAP UK

ADD.
MICHAEL WENZ, SAP AG
CHRISTIAN GEORGI, SAP AG
MARCO ERTEL, SAP AG
BETRAM GANZ, SAP AG
JEFF GEBO, SAP LABS.
PETER McNULTY, SAP LABS.

1. Exercise: Create an Order Approval Application for mobile devices

1.1 Objective

A manager needs to be able to reject or approve orders of customers by using a mobile connected device, e.g. a BlackBerry Wireless Handheld. An order consists of order items, a final amount (price) as well as a proposed discount. Approving or rejecting an order depends on the discount with respect to the items and the final price of the order.

1.2 Task

Create an application that fulfils the following requirements

- search customer orders
- display the search results in a list
- selecting orders from the list to display in a further view allowing order to be approved or rejected depending on the orders properties

1.3 Solution

Starting point

Within the SAP NetWeaver Developer Studio you find two projects (if not then see remark below):

- ***TutWD_Mobile_Init*** (Mobile Exercise)
a skeleton project containing the initial starting point for the exercise session
- ***TutWD_Mobile*** (Mobile Solution)
a fully functional and complete Web Dynpro project, containing both a minimal form factor application for mobile devices, as well as a big screen form factor application for desktop browsers; should be used as example and to see the results

Remark: If these projects do not appear in the SAP NetWeaver Developer Studio yet you probably have to

- copy the folders "TutWD_Mobile_Init" and "TutWD_Mobile" to your Laptop/PC
- import the two projects into the workspace of your SAP NetWeaver Developer Studio (File -> Import -> Existing Project into Workspace)
- if the rebuild (click right on project, Rebuild project) fails, "repair" the project (right click on project, Repair -> Project Structure and Classpath)

Additional Information on the *TUTWD_MOBILE_INIT* (Mobile Exercise) project

The project contains a single Web Dynpro Component called ***ApprovalComp***. This component consists of two entities:

- ComponentController ***ApprovalComp***: provides data management functionality, such as implementing the search algorithm, as well as providing the searchResult via a contextNode for use by views, which can map their contexts to the component controller.
- Window ***ApprovalWindow***: This window is initially empty (contains no views). Any views defined by the app developer can be arranged in this window.

Step 1

Define a view for searching called *SearchView*, a view for displaying the searchResults called *ResultView* as well as defining an application so that the view can be displayed in the Browser/emulator:

- Go to the *TutWD_Mobile_Init* project.
- Open the Data Modeller by double clicking on the component *ApprovalComp*.
 - o Create a view *SearchView*.
 - o Create a datalink from *SearchView* to *ComponentController*.
 - o In the wizard, drag and drop the context attribute *searchCustomerName* from the *ApprovalComp* to the left side *SearchView* and select the new attribute in the next dialog. Press **OK** and **Finish**. Edit the View *SearchView*:
 - o Go to the **Layout** Tab of the *View Designer*.
 - o Delete the default TextView UI Element.
 - o In the Outline, add a **Group** UIElement (insertChild on the RootUIElementcontainer).
 - o The Group UI Element has a child UIElement Caption (Group1_Header) which contains the header information of the group. Go to the properties of this header and set the *text* property to *Search for Orders*.
 - o In the Outline area click right on the Group UIElement and select **Apply Template**. In the wizard, select **Form**, followed by **next**.
 - o Select the only available contextAttribute *searchCustomerName* and select **finish**. Notice how the view now contains a label and an InputField.
 - o In the Outline area again select **Apply Template**. In the wizard, select **ActionButton**, **next**.
 - o As label text, enter *Search*, and click **next**.
 - o In the section **Invoke Methods**, select **Call Method**. From the List, select the component controller *ApprovalComp* and the method *searchOrders*. This will cause the method *searchOrders* to be called, whenever the search button is pressed.
 - o In the section **Fire Plug**, select **Fire Plug**. Press the button **New** on the right hand side to create a new outbound plug. Enter *ToResultView* and press **ok**.
 - o Press **finish**.

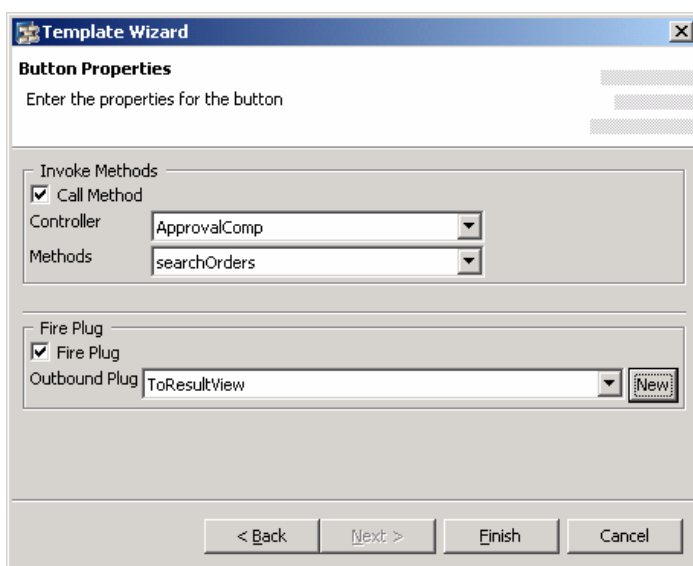


Figure 1: Template Wizard while creating a button

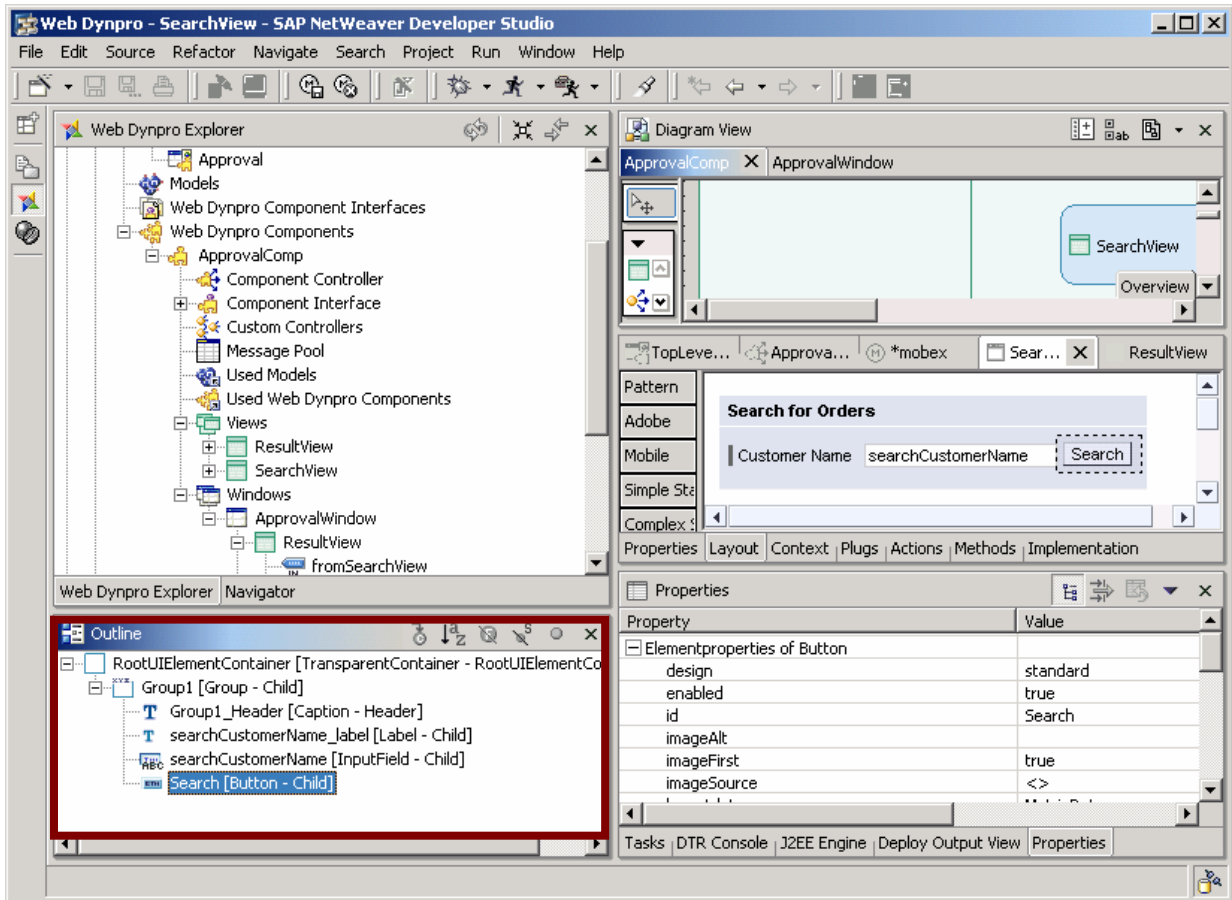


Figure 18: Outline – Button needs to be part of Group1

1. Make sure the button is contained within the group.

- o Make sure the button is contained within the group. If this is not the case, you can move it there using the outline.
- o Go to the properties of UI Element Group and change the layout from *MatrixLayout* to *GridLayout* and set *colCount* to 3. This is essential since the BlackBerry does not understand MatrixLayout.

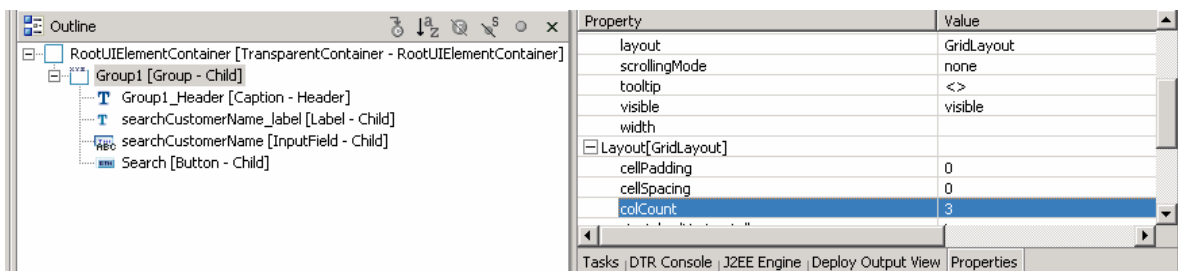


Figure 19: Properties of Group 1: GridLayout, colCount 3

- Add the *SearchView* to the *ApprovalWindow*, so that it will be displayed:
 - o Click double on the window *ApprovalWindow*, the Navigation Modeller appears.

- In the palette, select the icon *embedView* and drop it anywhere in the diagram view.
 - In the wizard dialog, enter *Embed existing View*, press *Next* and select *SearchView* from the list.
 - Press *finish*.
 - In the palette, select the icon *inbound plug* and drop it on the newly created *SearchView*.
 - In the dialog, enter the plugs name: *startSearch* and press *finish*.
- Create a Web Dynpro application representing the URL to start your Mobile scenario
 - In the Web Dynpro Explorer, click right on *Applications*. Enter *Approval* as the application name in the package *sap.com.test.app*. Press *next*.
 - Select use existing component. Press next.
 - As *WebDynpro Component*, select *ApprovalComp*.
 - As *Interface View*, select *ApprovalWindowInterfaceView*.
 - As *Startup Plug*, select *Default*.

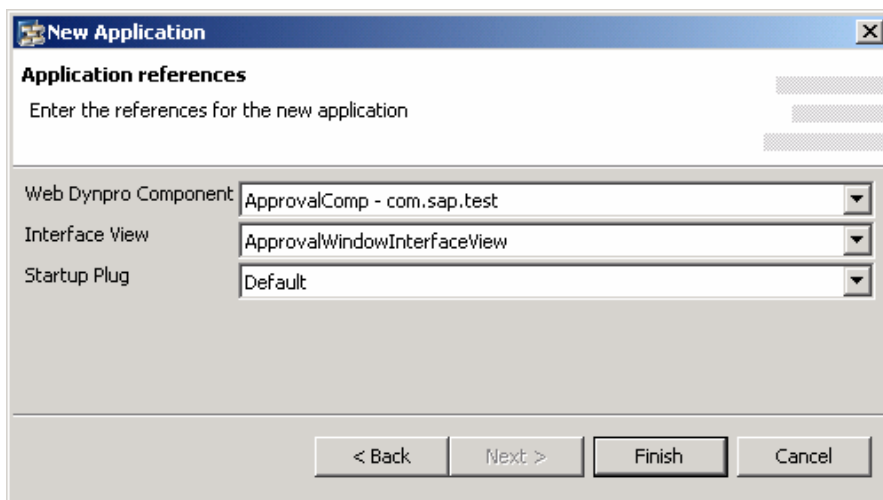
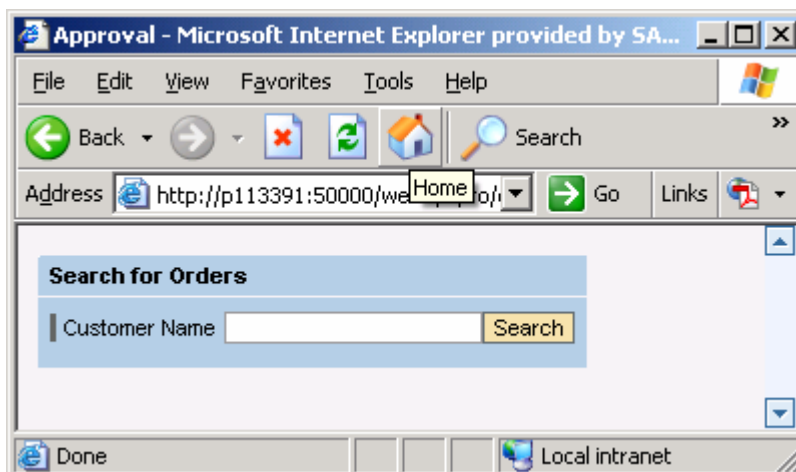


Figure 3: new application

- Press *finish*.
 - You are now ready to test the application!
- Test the application
 - In the Web Dynpro Explorer, click right on the application *Approval* and select *Deploy new Archive and Run*.
 - This will start the application in the full screen browser directly from the IDE.



Step 2

- Create a second view for displaying the search results called **ResultView**:
 - o Go to the Data Modeller (double click on ApprovalComp) and create the view **ResultView**.
 - o Create a data link from the **ResultView** to **ComponentController**.
 - o In the wizard, drag and drop the context node **ResultTable** from the **ApprovalComp** to the left side **ResultView**.
 - o Select the node **ResultTable** on the left side in the Wizard Dialog. If you expand the node, make sure the child node **ItemsTable** as well as the child attributes of **ItemsTable** are NOT selected. Press **OK** and **Finish**.

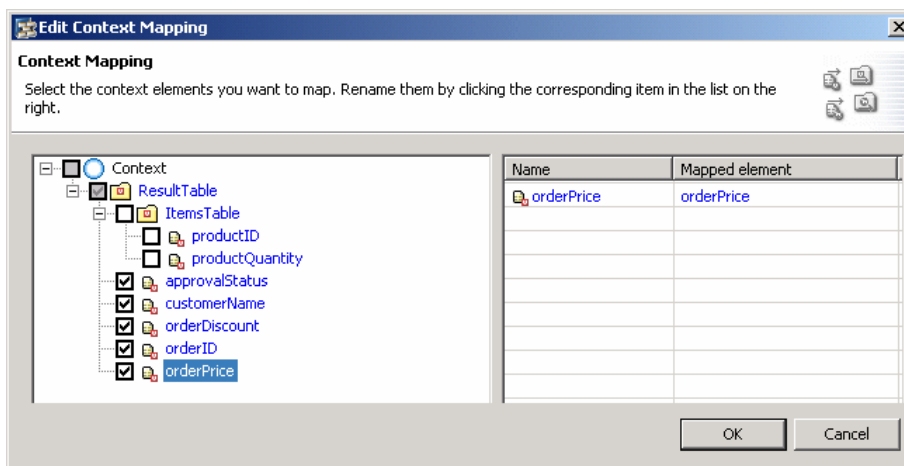


Figure 4: Context Mapping

- Edit the View **ResultView**:
 - o Go to the **Layout** tab.
 - o Delete the default Text View UI Element.
 - o In the outline, add a **TransparentContainer** UIElement (insertChild on the RootUIElementcontainer).
 - o Go to the properties of the **TransparentContainer** UIElement and set the **layout** property to **GridLayout**.
 - o Click right on the **TransparentContainer** UIElement and select **Apply Template**. In the wizard, select **Table**, followed by **next**.
 - o Select the contextAttributes **customerName**, **orderID**, and **approvalStatus** and select **next**.

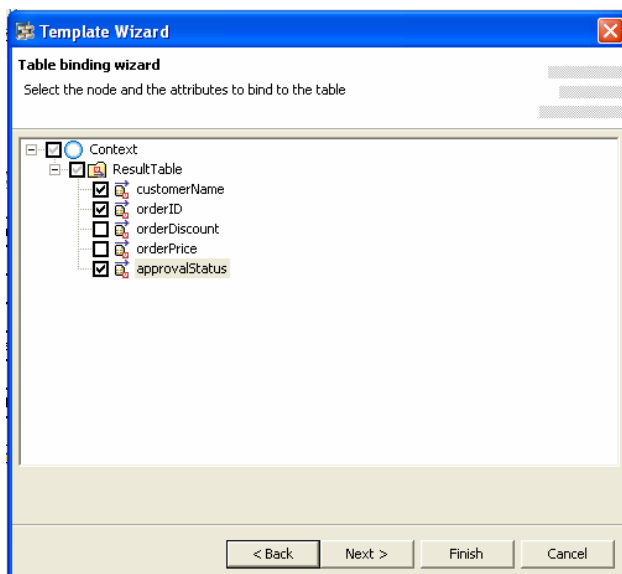


Figure 5: Template wizard while table binding

- o In the Editor column, change the row *approvalStatus* from *DropDownByKey* to *TextView*. This is needed because the approval status should not be changed in this view!! Press *finish*.

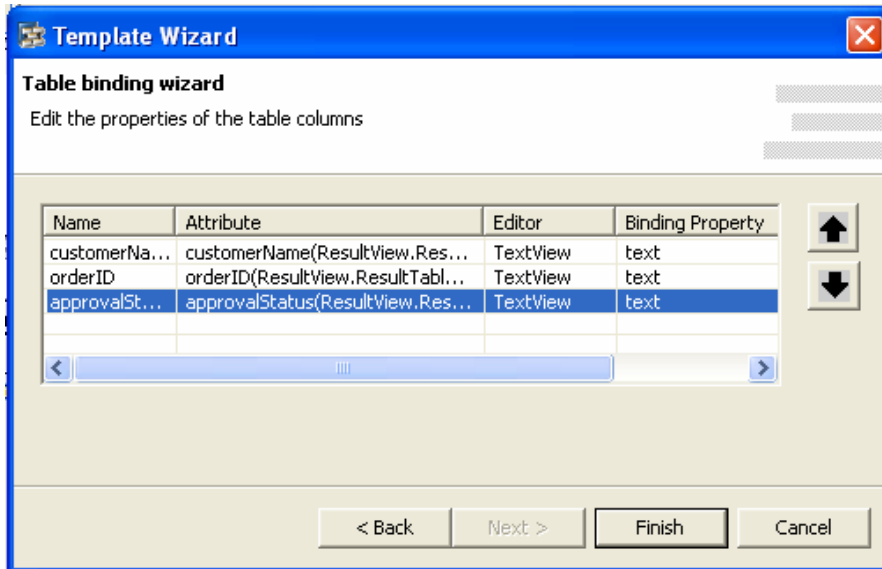


Figure 6: Template wizard while table binding

- o Notice how the view now contains a table with three columns.

Name	ID	Appr. status
ResultTable.customerName	ResultTable.orderID	ResultTable.approvalStatus
ResultTable.customerName	ResultTable.orderID	ResultTable.approvalStatus
ResultTable.customerName	ResultTable.orderID	ResultTable.approvalStatus
ResultTable.customerName	ResultTable.orderID	ResultTable.approvalStatus
ResultTable.customerName	ResultTable.orderID	ResultTable.approvalStatus

Figure 7: layout of the ResultView

- o Click right on the table UIElement within the outline area and select *Insert Header*.
- o Select the property *text* of the newly created header (Caption UIElement). Set the text value to *Result Of Search:*.
- o Click right on the *TransparentContainer* UIElement and select *Apply Template*. In the wizard, select *Action Button*, followed by *next*.
- o As label text, enter *Restart*, and click *next*.
- o In the section *Fire Plug*, select *Fire Plug*. Press the button *New* on the right hand side to create a new outbound plug. Enter *ToSearchView* and press *OK*.
- o Press *Finish*.

- Add the **ResultView** to the **ApprovalWindow**, so that it will be used in the navigation graph:
 - o Click double on the window **ApprovalWindow**.
 - o In the palette, select the icon **embedView** and drop it anywhere in the diagram view.
 - o In the WizardDialog, enter **Embed existing View**, press **Next** and select **ResultView** from the list.
 - o Press **finish**. The navigation modeller looks like this. Both outbound plugs exist and one inbound plug exists.

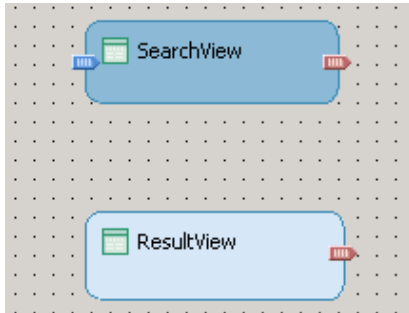


Figure 8: navigation modeller

- o In the palette, select the icon **inbound Plug** and drop it on the newly created **ResultView**.
- o In the dialog, enter the plugs name: **fromSearchView** and press **finish**.
- o In the palette, select the icon **create a navigation Link**. From the **SearchView** connect the outbound plug **ToResultView** with the inbound plug **fromSearchView** on the **ResultView**.
- o In the palette, select the icon **create a navigation Link**. From the **ResultView** connect the outbound plug **ToSearchView** with the inbound plug **startSearch** on the **SearchView**.

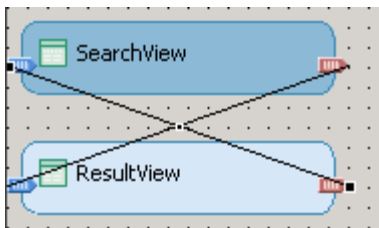


Figure 9: Navigation between SearchView and ResultView

- Test the application
 - o In the Project Browser, click right on the application **Approval** and select **Deploy new Archive and Run**.
 - o This will start the application in the full screen browser directly from the IDE.
 - o Click on the **Search** Button.

Search for Orders

Customer Name

Figure 10: Browser: SearchView

Result of Search			
	Name	ID	Appr. status
<input type="checkbox"/>	Customer01	Order01-001	?
<input type="checkbox"/>	Customer02	Order02-001	?
<input type="checkbox"/>	Customer02	Order02-002	?
<input type="checkbox"/>	Customer03	Order03-001	?
<input type="checkbox"/>	Customer03	Order03-002	?

1 von 13

Figure 11: Browser: ResultView

- o Type in the URL into the BlackBerry Simulator to see the application on a mobile device and click on Search.

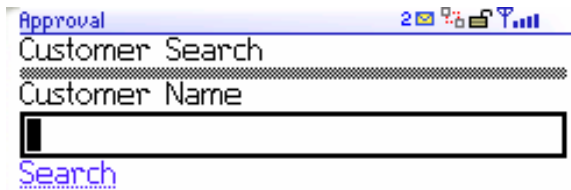
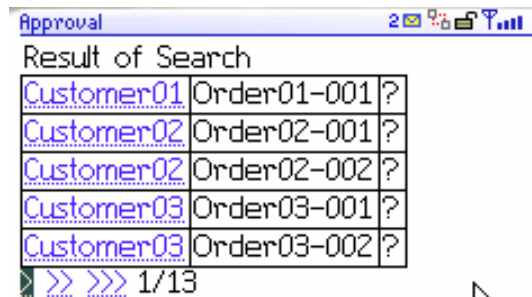


Figure 12: BlackBerry: SearchView



[Restart](#)

Figure 13: BlackBerry: ResultView

Remark: Notice that the search results are now displayed. The search does a pattern match, therefore, entering, e.g. "1" will return 2 lines, and entering nothing will return everything.

Remark: Notice that the approval status always contains a question mark. The data provided only allows the following entries "?" for status unknown, "A" for approved, "R" for rejected. This is not suitable for display purposes. The next step will describe how this technical representation can be converted to a human readable approval status.

Remark: Don't worry if in some cases icons are not properly displayed on the BlackBerry Simulator. This is an issue of the release of the browser on the Simulator.

Step 3:

Use a calculated context attribute, to properly display the approval status in a human readable way.

- Edit the View **ResultView**:
 - o Go to the **Context** tab.
 - o Click right on the node **ResultTable** of the **ViewContext** (left pane).
 - o Select **New** -> **ValueAttribute**, enter **approvalDisplay** and press **finish**.
 - o Select the newly created ValueAttribute **approvalDisplay** and display the properties.
 - o Set the property **readonly** to true.
 - o Set the property **calculated** to true. Notice that a new property **calculatedAttributeGetter** appears with the value **getResultTableApprovalDisplay**. This is the name of the method provided for implementing application specific code for returning the display value of this attribute.

Property	Value
calculated	true
calculatedAttributeGetter	getResultTableApprovalDisplay
name	approvalDisplay
readOnly	true
structureElement	
type	string

- o Switch to the **implementation** tab and go to the method **getResultTableApprovalDisplay**.
- o Place the following code between the @@begin and @@end comment lines:

```

if(element.getApprovalStatus().equals("R"))
    return "rejected !";
else if(element.getApprovalStatus().equals("A"))
    return "approved !";

```

```
return "tbd...";
```

- o Alternatively, you can adapt the above code to return whatever you feel suitable for display.
- o Switch to the *Layout* tab.
- o Click right on the header line of the *Table* UIElement and select *Create Binding*.
- o Select the contextAttribute *approvalDisplay*. Leave the other attributes selected and press *Next* and *finish*.

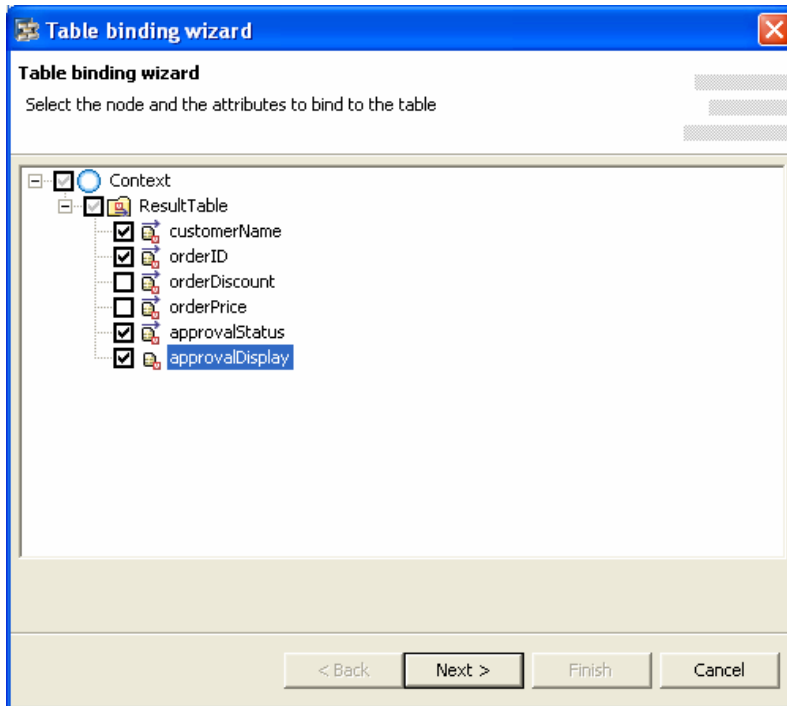


Figure 14: Table binding wizard

- o Now delete the TableColumn *approvalStatus* in the Outline area of the SAP NetWeaver Developer Studio.
- Test the application again.

Step 4:

Define a view for approving the currently selected order called *ApprovalView*.

- Data Modeller (double click on component *ApprovalComp*):
 - o Create view *ApprovalView*.
 - o Create a data link from *ApprovalView* to *ComponentController*.
 - o In the wizard, drag and drop the context node *ResultTable* from the *ApprovalComp* to the left side *ApprovalView*. Expand the node *ResultTable* on the left side in the Wizard Dialog. Select everything. Press *OK* and *Finish*.
- Edit the View *ApprovalView*:
 - o Go to the *Layout* tab.
 - o Delete the default TextView UI Element.
 - o In the outline, add a *TransparentContainer* UIElement (insertChild on the RootUIElementcontainer).
 - o Go to the properties of the *TransparentContainer* UIElement and set the *layout* property to *GridLayout*.
 - o Click right on the *TransparentContainer* UIElement and add a *Group* UIElement (insert Child on the RootUIElementcontainer).
 - o The Group UIElement will have a child UIElement which contains the header information of the group. Go to the properties of this header and set the *text* property to *Order Approval*.

- Click right on the Group UIElement and select **Apply Template**. In the wizard, select **Form**, followed by **next**.
- Select the following contextAttributes of the node ResultTable: **customerName**, **orderID**, **orderDiscount**, **orderPrice**. Select **next**.
- In the editor column, change all rows from **InputField** to **TextView**. This is needed because this data is not to be manipulated in this view!! Press **finish**.

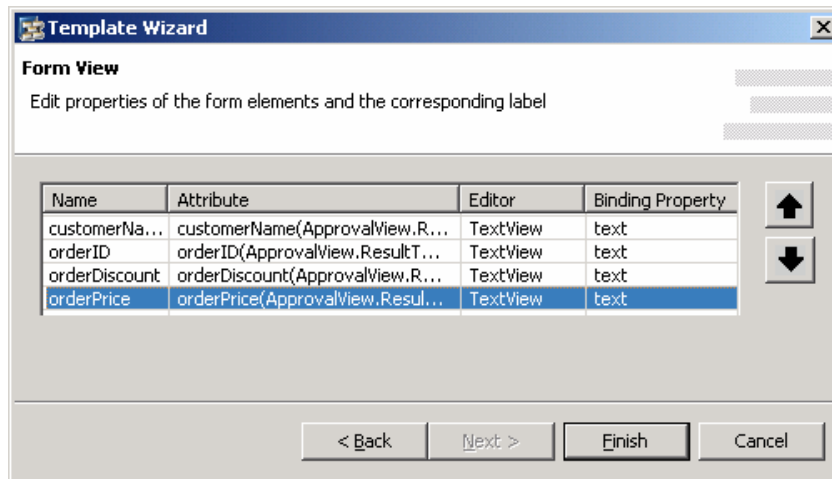


Figure 15: Form template

- Change the **layout** property of the UI Element Group from **MatrixLayout** to **GridLayout** and change the **colCount** to 2.
- Click right on the **TransparentContainer** UIElement and select **Apply Template**. In the wizard, select **Table**, followed by **next**.
- Select the contextAttributes **productId** and **productQuantity** (both attributes are part of the Node **ItemsTable**.) Select **next** and **finish**.

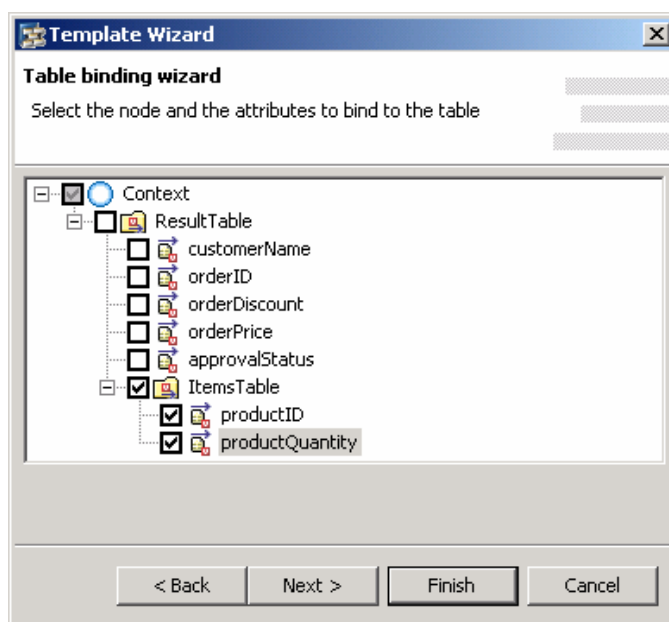


Figure 16: Table binding wizard

- Click right on the Table UIElement and select **Insert Header**.
- Select the property **text** of the newly created Header (Caption UIElement). Set the value to **Order Items**.

- RightClick on the *TransparentContainer* UIElement and select *Apply Template*. In the wizard, select *Action Button*, followed by *next*.
- As label text, enter *Approve*, and click *next*.
- In the section *Fire Plug*, select *Fire Plug*. Press the button *New* on the right hand side to create a new outbound plug. Enter *ToResultView* and press *OK*.
- Press *finish*.
- Click right on the *TransparentContainer* UIElement and select *Apply Template*. In the wizard, select *Action Button*, followed by *next*.
- As label text, enter *Reject*, and click *next*.
- In the section *Fire Plug*, select *Fire Plug*. Select the plug *ToResultView* from the list and press *OK*.
- Press *finish*.
- Click right on the *TransparentContainer* UIElement and select *Apply Template*. In the wizard, select *Action Button*, followed by *next*.
- As label text, enter *Restart*, and click *next*.
- In the section *Fire Plug*, select *Fire Plug*. Press the button *New* on the right hand side to create a new outbound plug. Enter *ToSearchView* and press *OK*.
- Press *finish*.
- Now, the approval and reject actions must be implemented to change the value of the appropriate data element.
- Switch to the *implementation* tab and go to the action eventhandler *onActionApprove*.
- Place the following code after the @@begin comment line:

```
wdContext.currentResultTableElement().setApprovalStatus("A");
```

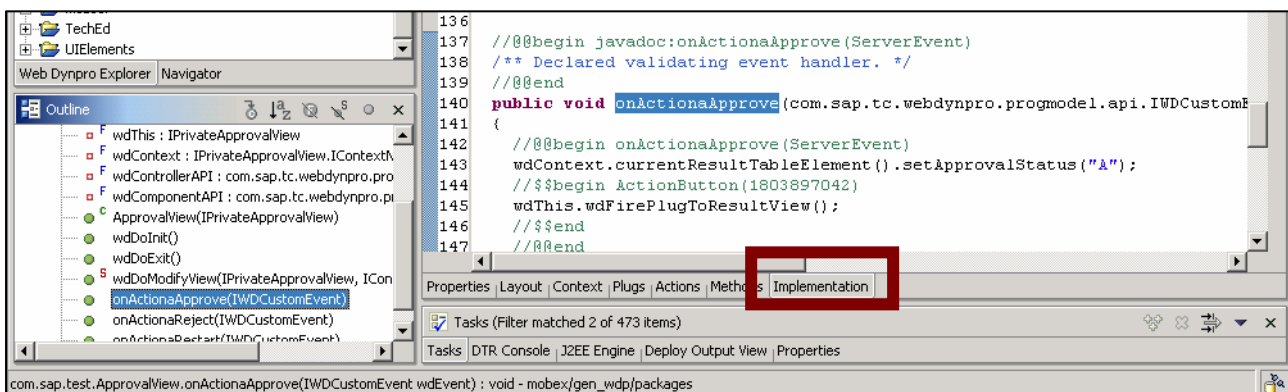


Figure 17: Implementation

- Go to the action eventhandler *onActionReject*
- Place the following code after the @@begin comment line:

```
wdContext.currentResultTableElement().setApprovalStatus("R");
```

- Add the *ApprovalView* to the *ApprovalWindow*, so that it will be used in the navigation graph:
 - o Double-Click on the Window *ApprovalWindow*.
 - o In the palette, select the icon *embedView* and drop it anywhere in the diagram view.
 - o In the WizardDialog, enter *Embed existing View*, press *Next* and select *ApprovalView* from the list.
 - o Press *finish*.
- o In the palette, select the icon *inbound Plug* and drop it on the newly created *ApprovalView*.
- o In the dialog, enter the plugs name: *fromResultView* and press *finish*.
- o In the palette, select the icon *create a navigation Link*. From the *ApprovalView* connect the Outbound Plug *ToSearchView* with the Inbound Plug *startSearch* on the *SearchView*.
- o In the palette, select the icon *outbound Plug* and drop it on the *ResultView*.
- o In the dialog, enter the plugs name: *ToApprovalView* and press *finish*.
- o In the palette, select the icon *create a navigation Link*. From the *ResultView* connect the Outbound Plug *ToApprovalView* with the Inbound Plug *fromResultView* on the *ApprovalView*.
- o In the palette, select the icon *create a navigation Link*. From the *ApprovalView* connect the Outbound Plug *ToResultView* with the Inbound Plug *fromSearchView* on the *ResultView*.

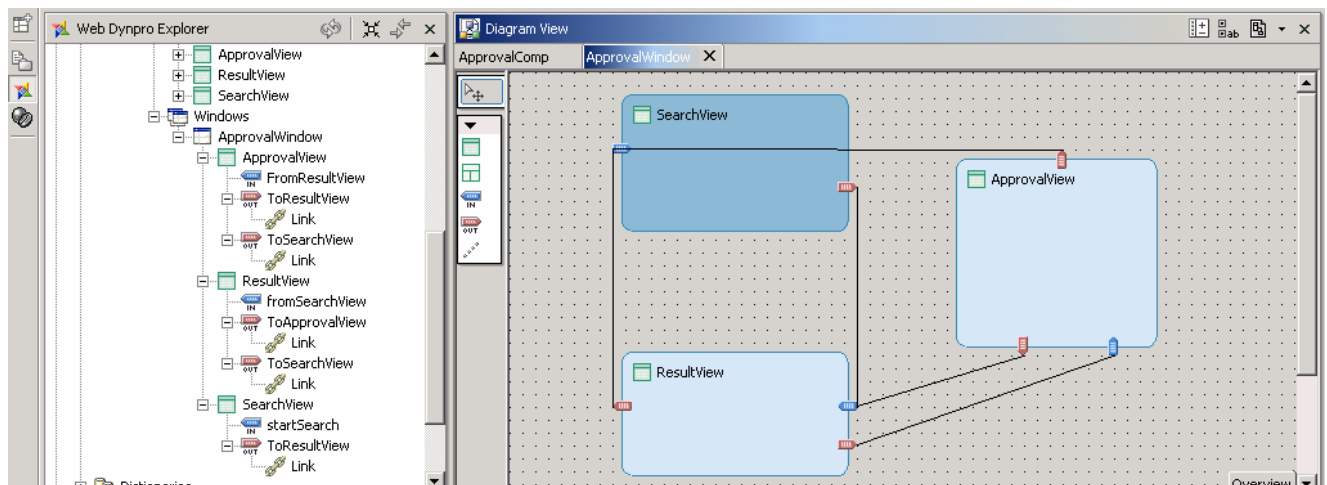


Figure 18: all the three views with the navigation links

- Edit the View **ResultView** (Up to now, no action has been defined to fire the plug to the ApprovalView once a new line in the table is selected):
 - o Go to the **Layout** tab.
 - o Click on the Table UIElement and select the property **onLeadSelect** in the **Events** section..
 - o Press the button “...” to open the dialog for creating an action.
 - o Enter **LineSelection** in the name field and **select new line** in the text field.
 - o In the **Fire Plug** drop down, select the plug **ToApprovalView** and press **finish**.

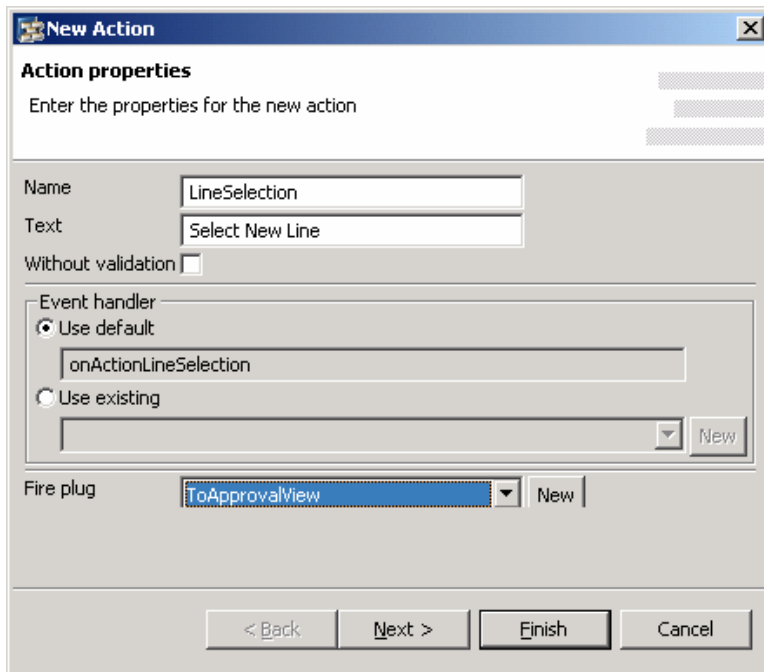
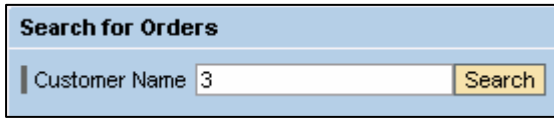


Figure 19: new action

- Test the application
 - o In the Project Browser, right-click on the application *Approval* and select *Deploy new Archive and Run*.
 - o This will start the application in the full screen browser directly from the IDE.

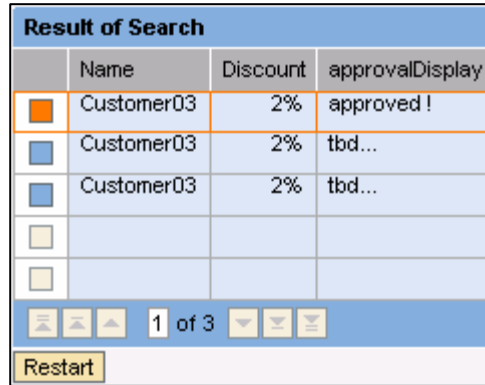
Remark: After searching, you can select one of the result items in the list, which displays the ApprovalView. Here you can select to approve or reject, which should be reflected in the data of the result list.



Search for Orders

Customer Name

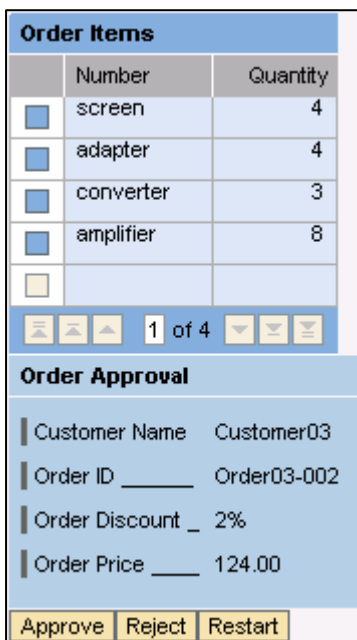
Figure 20: Browser: SearchView



	Name	Discount	approvalDisplay
<input checked="" type="checkbox"/>	Customer03	2%	approved !
<input type="checkbox"/>	Customer03	2%	tbd...
<input type="checkbox"/>	Customer03	2%	tbd...
<input type="checkbox"/>			
<input type="checkbox"/>			

1 of 3

Figure 21: Browser: ResultView



Order Items	
Number	Quantity
<input type="checkbox"/> screen	4
<input type="checkbox"/> adapter	4
<input type="checkbox"/> converter	3
<input type="checkbox"/> amplifier	8
<input type="checkbox"/>	

1 of 4

Order Approval

Customer Name Customer03

Order ID _____ Order03-002

Order Discount _ 2%

Order Price ____ 124.00

Figure 22: Browser: ApprovalView

First you choose the first row of the *Customer03*. You get Figure 39, approve the order and as a result you get Figure 38 with “approved” in the approvalDisplay.

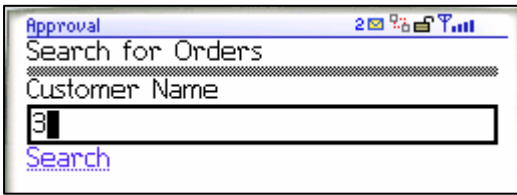


Figure 24: BlackBerry: SearchView

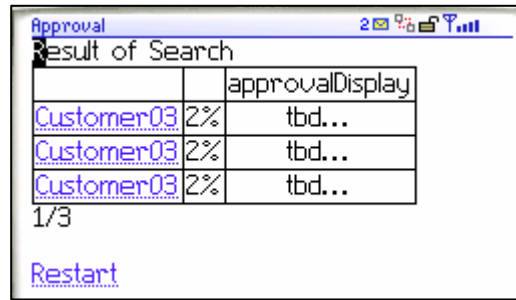


Figure 23: BlackBerry: ResultView 1

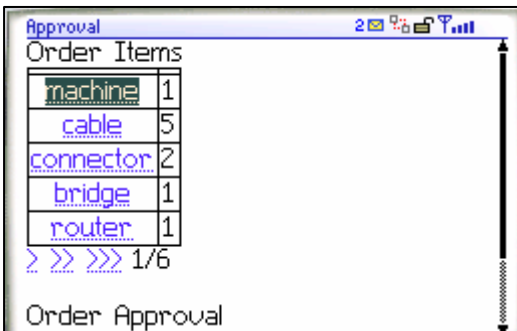


Figure 25: BlackBerry: ApprovalView

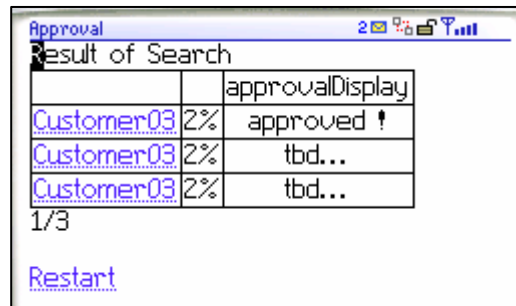


Figure 26: BlackBerry: ResultView 2

Copyright 2004 SAP AG. All Rights Reserved

- No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.
- Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.
- Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.
- IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.
- Oracle is a registered trademark of Oracle Corporation.
- UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.
- Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.
- HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.
- Java is a registered trademark of Sun Microsystems, Inc.
- JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.
- MaxDB is a trademark of MySQL AB, Sweden.
- SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.
- These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP assumes no responsibility for errors or omissions in these materials.