Working with the MultiPane UI Element in Web Dynpro ABAP

Applies to:

Web Dynpro ABAP

Summary

This tutorial shows the use of the MultiPane UI element in Web Dynpro ABAP applications.

Author: Uday Gubbala

Company: Cognizant Technology Solutions

Created on: 17th December 2008

Author Bio

Uday Gubbala is a senior development consultant working for Cognizant Technology Solutions India. He has been involved in various projects as a technical consultant in ABAP and Web Dynpro.

Table of Contents

Objective Exercise

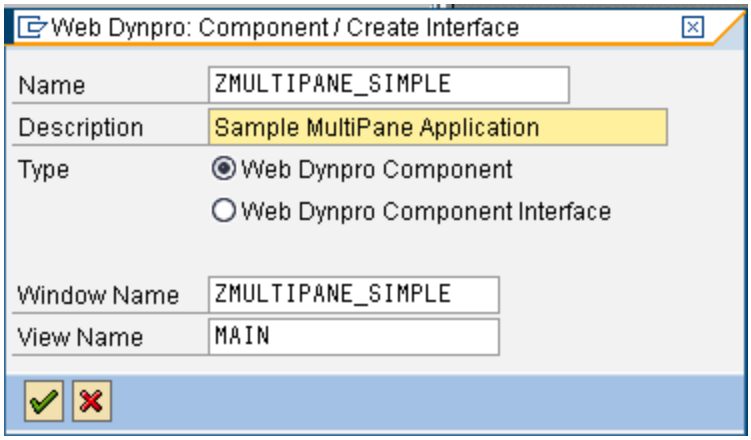
Create an application that uses a MultiPane to display the list of all purchase orders & their corresponding item details. The user will have to enter a vendor number in the InputField provided & press on enter. The MultiPane would then display the corresponding purchase orders & item details.

Prerequisites

Basic knowledge of programming in ABAP and Web Dynpro for ABAP is required.

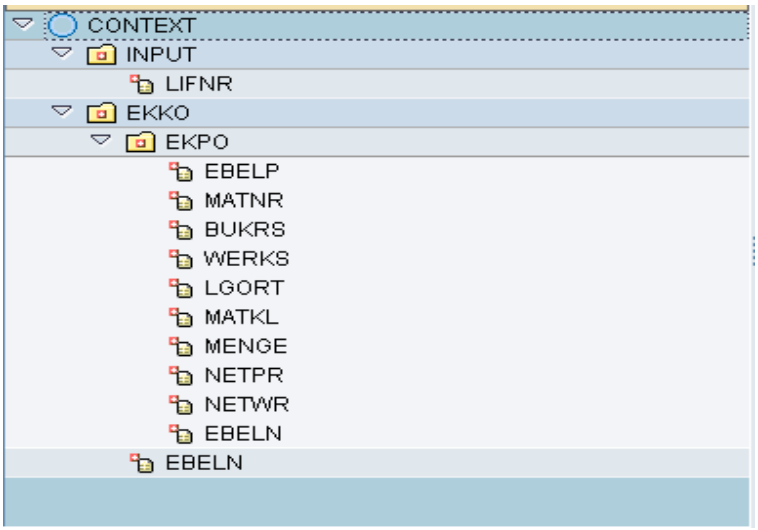
Create Web Dynpro component

To accomplish the objective of the exercise mentioned above, we will first create a new Web Dynpro component in SE80. Name it as ZROADMAP_SAMPLE.



Create context attributes in your MAIN view

Create a node in the context of the component controller. Name it as INPUT. Create an attribute LIFNR under this node. This is of the same type as the field LIFNR of the LFA1 table. We will use this field as the input to be taken from the user.
Create another node named EKKO. This node is of the type dictionary structure EKKO. Create an attribute EBELN under this node from the table EKKO. This node would serve as the source for holding the purchase order number value. Now create a sub-node for EKKO & name it as EKPO. Select the attributes to be added under this node from the table EKPO. This node would hold the purchase order item details information. Specify a supply function S_FILL_PO_ITEMS for EKPO. Your MAIN view's context should look like below:

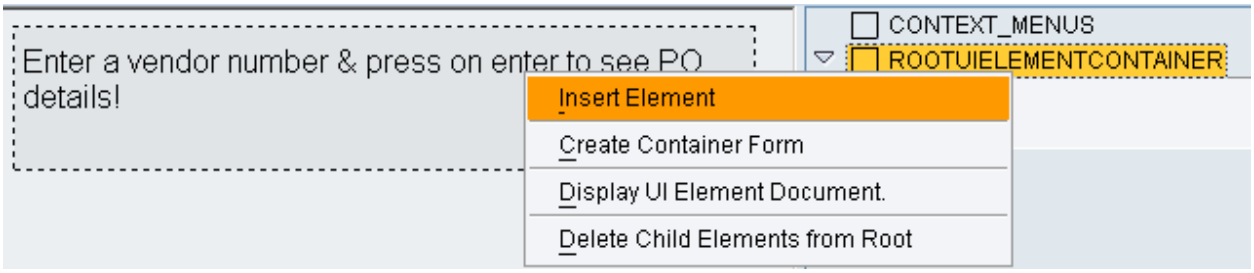


Define view layout

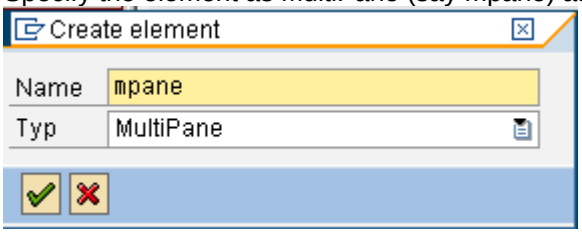
Place a caption & InputField on your view as how shown below:

Bind the "value" property of the InputField to the attribute LIFNR which we had created under the node INPUT. Also create an action "FETCH_DATA" for the event "onEnter" of the InputField. We would be coding our logic to fetch the purchase order & item details information in here later.

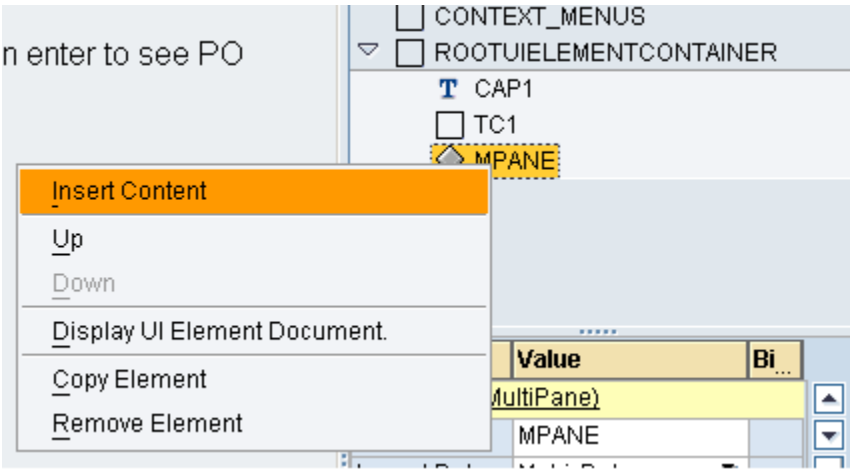
Now we will start designing the layout for incorporating the MultiPane ui element. Right click on the ROOTUIELEMENTCONTAINER element & say "Insert Element" as how shown below:



Specify the element as MultiPane (say mpane) as how shown below:



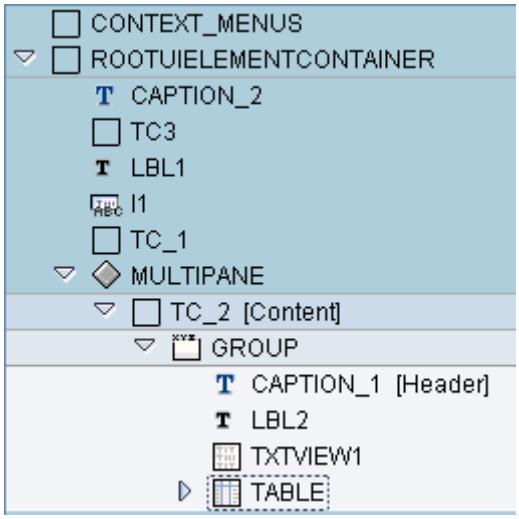
Now right click on your MultiPane element and say, "Insert Content" as how shown below.



Bind the "dataSource"property of MultiPane (mpane) to the context node EKKO.

Right click on MultiPane & Insert a TransparentContainer (say TC2) within the MultiPane element.
Now right click on TransparentContainer (TC2) & select “Insert Element”. Specify the element name as GR1 & the element type as Group. Specify the caption property for Group (GR1) as “Details”.

Now embed a label (say lbl1) & a TextView (say txtview1) within this group by right clicking on GR1 & choosing “Insert Element”. Specify the “text” property of the label lbl1 as “PO Number”. Bind the TextView (txtview1) to the attribute EBELN created under node EKKO. Right click on the group GR1 and select “Insert Element” to embed a table within this group. Bind the table to node EKPO & specify the property “text” of the table’s caption as “Items”. Now your ui hierarchy should look like shown below:



Am creating an attribute GV_MESSAGE_MANAGER of type ref to IF_WD_MESSAGE_MANAGER & within the WDDOINIT method am saving the message manager reference to re-use throughtout the program.

Implement methods of the MAIN view

```
METHOD wddoinit .
*   get message manager
  DATA lo_api_controller     TYPE REF TO if_wd_controller.
  DATA lo_message_manager    TYPE REF TO if_wd_message_manager.
  lo_api_controller ?= wd_this->wd_get_api( ).
  CALL METHOD lo_api_controller->get_message_manager
    RECEIVING
      message_manager = lo_message_manager.
** Saving the message manager reference to use throughtout the program
  wd_this->gv_message_manager = lo_message_manager.
ENDMETHOD.
```

Put the following coding into the action handler for the InputField’s onEnter event.

```
    METHOD onactionfetch_data .
      DATA: lr_node TYPE REF TO if_wd_context_node,
            lv_lifnr TYPE wd_this->element_input-lifnr,
            lt_ekko TYPE wd_this->elements_ekko,
            wa_ekko TYPE wd_this->element_ekko.

      lr_node = wd_context->get_child_node( name = if_main=>wdctx_input ).
      lr_node->get_attribute( EXPORTING name  = 'LIFNR'
```

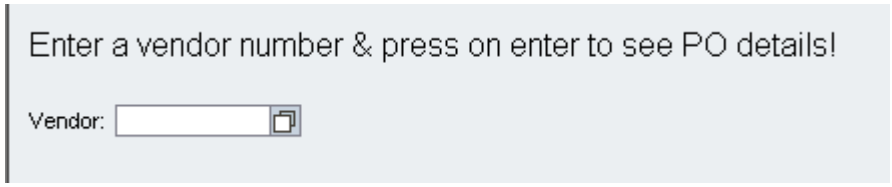
```
                              IMPORTING value = lv_lifnr ).
      SELECT SINGLE * FROM lfa1 INTO CORRESPONDING FIELDS OF wa_ekko WHERE lifnr =
    lv_lifnr.

      IF sy-subrc <> 0.
    ** Raise an error message that it's an invalid vendor number
        CALL METHOD wd_this->gv_message_manager->report_error_message
          EXPORTING
            message_text = 'No vendor exists by this number. Try again!'
    ** Clear the input field for the user to re-try
        lr_node->set_attribute( EXPORTING name  = 'LIFNR'
                                          value = '' ).

      ELSE.
        SELECT ebeln INTO TABLE lt_ekko FROM ekko WHERE lifnr = lv_lifnr.
        IF sy-subrc <> 0.
    **  Raise an error message if no PO exists for the entered vendor
          CALL METHOD wd_this->gv_message_manager->report_error_message
            EXPORTING
              message_text = 'No purchase orders exist for this vendor!'.
        ELSE.
    **  Get the PO details & bind them to node EKKO
          lr_node = wd_context->get_child_node( name = if_main=>wdctx_ekko ).
          lr_node->bind_table( new_items = lt_ekko ).
        ENDIF.
      ENDIF.
    ENDMETHOD.
```

Below is my coding for the supply function S_FILL_PO_ITEMS of node EKPO.

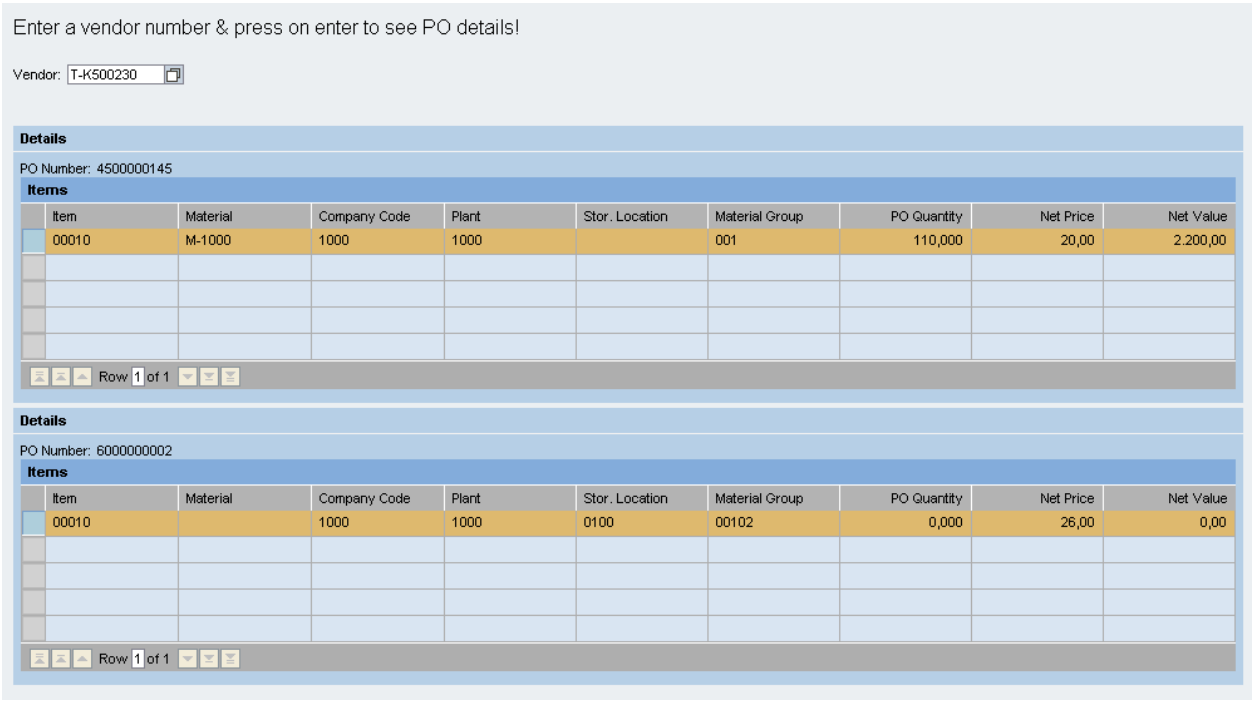
```
    METHOD s_fill_po_items .
      DATA: ls_parent_attributes TYPE wd_this->element_ekko,
            lt_ekpo TYPE wd_this->elements_ekpo,
            ls_ekpo LIKE LINE OF lt_ekpo.


** This would result in the ststem fetching the PO number value being displayed in the
current pane
  parent_element->get_static_attributes( IMPORTING static_attributes =
ls_parent_attributes ).


** Use the obtained current PO number to fetch relevant PO item details
  SELECT * FROM ekpo INTO CORRESPONDING FIELDS OF TABLE lt_ekpo WHERE ebeln =
ls_parent_attributes-ebeln.


  node->bind_table( new_items = lt_ekpo ).
ENDMETHOD.
```

Create and test the Web Dynpro application

Create an application for the component & try running the application. You would get an InputField prompting you to enter the vendor number like how shown below:



Now enter a vendor number & press enter. Since the dataSource property of the MultiPane is bound against the context node EKKO the system would create that many number of rows as the number of entries in the node EKKO. Within this node we have just 1 attribute EBELN so if the vendor entered by the user has 2 purchase orders then this MultiPane would replicate twice & as a result the output would be something like how shown below:



Related Content

Read more about the MultiPane element in the SAP library

For more information, visit the Web Dynpro for ABAP

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.