Conditionally Assigning Colors and Input Enable to ALV Columns in Web Dynpro ABAP

Applies to:
SAP Net Weaver 2004s. Web Dynpro for ABAP

Summary
This document explains how to assign colors and how to make columns editable conditionally of ALV in Web Dynpro for ABAP.

Author: Suman Kumar Chinnam
Company: YASH Technologies
Created on: 16 September 2008

Author Bio
Suman Kumar is a Consultant for YASH Technologies, Pvt., and Ltd. Based in India. He has 2 years of IT experience. He has good knowledge of ABAP, BSP and Web Dynpro for ABAP.
# Table of Contents

Create Columns Editable and assign colors to the cells ....................................................................................3

1. Add attributes to context .............................................................................................................................3

2. Create a Method to Populate the Data .......................................................................................................4

3. Populate Context attributes CELLDESIGN and READONLY and Bind the data to Context ................4

4. Make Column Editable and Set colors to column .......................................................................................5

Result ..................................................................................................................................................................6

Disclaimer and Liability Notice ............................................................................................................................7
Create Columns Editable and assign colors to the cells

In this document I am trying to show the Flights details using ALV. I will make some of the cells of columns editable and colored. To do this we have to take two context attributes one is to fill the color and other is to make the editable cells conditionally.

1. Add attributes to context

Add attributes READONLY of type ABAP_BOOL and CELLDESIGN of type WDUI_TABLE_CELL_DESIGN to the node.
2. Create a Method to Populate the Data

```
DATA: node_flights TYPE REF TO if wd_context_node,
  it_flights TYPE sflight tab1,
  ls_flights TYPE sflight,
  it_final TYPE if_componentcontroller=>elements_flights,
  ls_final TYPE if_componentcontroller=>element_flights.

SELECT * FROM sflight INTO TABLE it_flights
UP TO 100 ROWS.
node_flights = wd_context->get_child_node( 'FLIGHTS' ).
```

3. Populate Context attributes CELLDIVIDED and READONLY and Bind the data to Context

Based on price value I am assigning Celldesig and Readonly values.

```
LOOP AT it_flights INTO ls_flights.
  MOVE-CORRESPONDING ls_flights TO ls_final.
  IF ls_final-price = '185.00'.
    ls_final-readonly = abap_true.
    ls_final-celldivided = cl_wd_table_column=>e_cell_design-badvalue_light.
  ELSE.
    ls_final-readonly = ' '.
    ls_final-celldivided = cl_wd_table_column=>e_cell_design-goodvalue_light.
  ENDIF.
  APPEND ls_final TO it_final.
ENDLOOP.
```

Now bind the table to Context.

```
node_flights->bind_table(
  new_items = it_final
  set_initial_elements = abap_true
  * [INDEX] = INDEX
 ).
```
4. Make Column Editable and Set colors to column

To make column editable first we need to get the column reference from ALV for that create an instance of ALV. We will get Interface Controller reference then call the method GET_MODEL to get the table reference.

```
* Instantiate ALV
   DATA: l_ref_cmp_usage TYPE REF TO if_wd_component_usage.

   l_ref_cmp_usage = wd_this->wd_cpuse_alv().
   IF l_ref_cmp_usage->has_active_component() IS INITIAL.
   l_ref_cmp_usage->create_component().
   ENDIF.

   DATA: l_ref_interfacecontroller TYPE REF TO iwc1_salv_wd_table .
   l_ref_interfacecontroller = wd_this->wd_cpiicf_alv().
   DATA:
   l_value TYPE REF TO cl_salv_wd_config_table.
   * Call the method get_model to get the reference of table
   l_value = l_ref_interfacecontroller->get_model( ).

   * Make Price column editable
   DATA:L_COLUMN TYPE REF TO cl_salv_wd_column,
   L_COLUMN1 TYPE REF TO cl_salv_wd_column,
   lr_input TYPE REF TO cl_salv_wd_ue_input_field,
   L_input TYPE REF TO cl_salv_wd_ue_input_field.

   * Get Column reference
   L_COLUMN = l_value->if_salv_wd_column_settings->get_column( 'PRICE' ).

   * Create input field
     CREATE OBJECT lr_input
     EXPORTING
     value_fieldname = 'PRICE'

   * Make column editable
     L_COLUMN->set_cell_editor( value = lr_input ).

   * To make some cells non editable
   * Assign attribute READONLY to the Input
     lr_input->set_read_only_fieldname( value = 'READONLY' ).

     L_VALUE->if_salv_wd_column_settings->delete_column( id = 'READONLY' ).

   * Set colors to cells
   * Get Column reference
     L_COLUMN1 = L_VALUE->if_salv_wd_column_settings->get_column( 'CARRID' ).

   * Assign attribute CELLDISPLAY to the column
   * Now the column shows 2 colors
     L_COLUMN1->set_cell_design_fieldname( value = 'CELDISPLAY' ).

     L_VALUE->if_salv_wd_column_settings->delete_column( id = 'CELDISPLAY' ).
```
Result

In the below output Price column is editable and Airline column is colored conditionally.

If Price value is 185 then I made those cells input disabled and changed the color of Airline cells to red.
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.