

Dynamically Make the Columns Editable/Non-Editable on an Action of Cell Edit in ALV



Applies to:

SAP Net Weaver 2004s, Web Dynpro for ABAP. For more information, visit the [User Interface Technology homepage](#).

Summary

This Article explains how to make the some columns editable and some are non editable when the on an action of cell edit.

Author: Suman Kumar Chinnam

Company: YASH Technologies

Created on: 22 December 2008

Author Bio

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

Table of Contents

Introduction	3
Output	8
Related Content.....	8
Disclaimer and Liability Notice.....	9

Introduction

In my previous Articles/Wiki's I have explained how to make conditionally edit and color the cells of an ALV and How to make conditionally entire row is editable.

<https://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/f0625002-596c-2b10-46af-91cb31b71393>

<https://wiki.sdn.sap.com/wiki/x/QIDWAw>

In this Article I am going to explain. How to change the some columns editable and some are non editable on an action of cell edit.

To do this initially I am making the entire table is editable for reference follow the logic in the mentioned Wiki above. Then when I change the value and press enter of a cell in one particular column, the changed cell of that row should be editable. But the columns before the changed cell column and itself are non editable and columns after the changed cell are editable.

Check the below screen how the application looks like initially. All rows are editable.

View [Standard View] Print Version Export Check Append Row Insert Row Delete Row											
ID	No.	Date	Airfare	Curr.	Plane	Capacity in economy class	Occupied economy class	Booking total	Capacity business class	Occupied	
AA	0017	27.02.2008	422.94	USD	747-400	385	357	193,812.47	31		
AA	0017	26.03.2008	422.94	USD	747-400	385	354	191,579.41	31		
AA	0017	23.04.2008	422.94	USD	747-400	385	356	193,846.29	31		
AA	0017	21.05.2008	422.94	USD	747-400	385	353	191,025.21	31		
AA	0017	18.06.2008	422.94	USD	747-400	385	349	192,230.75	31		
AA	0017	16.07.2008	422.94	USD	747-400	385	355	192,061.46	31		
AA	0017	13.08.2008	422.94	USD	747-400	385	355	194,129.57	31		
AA	0017	10.09.2008	422.94	USD	747-400	385	356	193,888.50	31		
AA	0017	08.10.2008	422.94	USD	747-400	385	353	192,010.80	31		
AA	0017	05.11.2008	422.94	USD	747-400	385	189	108,361.60	31		
AA	0017	03.12.2008	422.94	USD	747-400	385	37	29,191.34	31		
AA	0017	31.12.2008	422.94	USD	747-400	385	115	68,123.03	31		
AA	0017	28.01.2009	422.94	USD	747-400	385	30	24,864.72	31		
AA	0017	25.02.2009	422.94	USD	747-400	385	37	29,060.21	31		
AA	0017	25.03.2009	422.94	USD	747-400	385	5	11,677.39	31		
AA	0064	29.02.2008	422.94	USD	A310-300	280	253	135,294.41	22		
AA	0064	28.03.2008	422.94	USD	A310-300	280	253	135,053.37	22		
AA	0064	25.04.2008	422.94	USD	A310-300	280	253	134,380.84	22		
AA	0064	23.05.2008	422.94	USD	A310-300	280	247	131,166.55	22		
AA	0064	20.06.2008	422.94	USD	A310-300	280	247	130,299.55	22		
AA	0064	18.07.2008	422.94	USD	A310-300	280	253	134,169.42	22		
AA	0064	15.08.2008	422.94	USD	A310-300	280	253	133,539.19	22		
AA	0064	12.09.2008	422.94	USD	A310-300	280	248	130,430.56	22		

Now If I change the value in Airfare and press enter I want to make first 5 columns are non editable including the Airfare column and remaining columns are editable. But the changed value in cell row is editable.

Here I am implementing one event handler method ON_ACTION_ENTER of event type ON_DATA_CHECK of interface controller (ALV) to handle the event 'Enter', when user changed the value and press enter in the cell. Write the code shown below to get the required output.

Method ON_ACTION_ENTER.

```

data:
  lo_nd_flights type ref to if_wd_context_node,
  lt_flights   type if_main=>elements_flights,
  ls_flights   type if_main=>element_flights,
  ls_final     type if_main=>element_flights,
  lt_final     type if_main=>elements_flights,
  ls_mod_cell  type SALV_WD_S_TABLE_MOD_CELL,
  lv_tabix     type sy-tabix.

* navigate from <CONTEXT> to <FLIGHTS> via lead selection
lo_nd_flights = wd_context->get_child_node( name = wd_this->wdctx_flights ).

* Get the table data
call method lo_nd_flights->get_static_attributes_table
  importing
    table = lt_flights .

* Populate READ_ONLY attribute
* The changed cell row is editable and remaining are non editable
read table r_param->T_MODIFIED_CELLS into ls_mod_cell
  with key attribute = 'PRICE'.

if sy-subrc eq 0.
  loop at lt_flights into ls_flights.
    lv_tabix1 = sy-tabix.
    if lv_tabix = ls_mod_cell-index.
      ls_flights-read_only = abap_false.
    else.
      ls_flights-read_only = abap_true.
    endif.
    modify lt_flights from ls_flights index
      lv_tabix1 transporting read_only.
  endloop.
endif.

* Bind internal table to context node
call method lo_nd_flights->bind_table
  exporting
    new_items      = lt_flights
*   set_initial_elements = ABAP_TRUE
*   index          =
  .

* Instantiate ALV and call GET_MODEL method
data lo_cmp_usage type ref to if_wd_component_usage.

lo_cmp_usage = wd_this->wd_cpuse_alv( ).
if lo_cmp_usage->has_active_component( ) is initial.

```

```

    lo_cmp_usage->create_component( ).
endif.

data lo_interfacecontroller type ref to iwci_salv_wd_table .
lo_interfacecontroller = wd_this->wd_cpifc_alv( ).

data lo_value type ref to cl_salv_wd_config_table.
lo_value = lo_interfacecontroller->get_model(
).

data:
lt_columns type SALV_WD_T_COLUMN_REF,
ls_columns type SALV_WD_S_COLUMN_REF,
lr_input type ref to CL_SALV_WD_UIE_INPUT_FIELD,
lr_text type ref to CL_SALV_WD_UIE_TEXT_VIEW,
lr_column type ref to cl_salv_wd_column,
lt_node_info type WDR_CONTEXT_ATTR_INFO_MAP,
ls_node_info type WDR_CONTEXT_ATTRIBUTE_INFO,
lv_tabix type sy-tabix,
lr_info type ref to IF_WD_CONTEXT_NODE_INFO.

types:
begin of ty_name,
name type string,
end of ty_name.

data:
lt_name type table of ty_name,
ls_name type ty_name.

lr_info = lo_nd_flights->get_node_info( ).
lt_node_info = lr_info->get_attributes( ).

loop at lt_node_info into ls_node_info.
ls_name-name = ls_node_info-name.
append ls_name to lt_name.
endloop.

call method lo_value->if_salv_wd_table_settings~set_visible_row_count
exporting
value = 50
.

* Get all the columns references
call method lo_value->if_salv_wd_column_settings~get_columns
receiving
value = lt_columns .

* Make the first five column editable and remaining are non editable
loop at lt_columns into ls_columns.
lr_column = ls_columns-r_column.
lv_tabix = sy-tabix.

```

```
read table lt_name into ls_name index lv_tabix.
case ls_name-name.
when 'CARRID'.
  create object lr_input
    exporting
      value_fieldname = ls_name-name.

  call method lr_column->set_cell_editor
    exporting
      value = lr_input .
  call method lr_input->set_read_only
    exporting
      value = ABAP_TRUE
  .

when 'CONNID'.
  create object lr_input
    exporting
      value_fieldname = ls_name-name.
  call method lr_column->set_cell_editor
    exporting
      value = lr_input .
  call method lr_input->set_read_only
    exporting
      value = ABAP_TRUE
  .

when 'FLDATE'.
  create object lr_input
    exporting
      value_fieldname = ls_name-name.

  call method lr_column->set_cell_editor
    exporting
      value = lr_input .
  call method lr_input->set_read_only
    exporting
      value = ABAP_TRUE
  .

when 'PRICE'.
  create object lr_input
    exporting
      value_fieldname = ls_name-name.
  call method lr_column->set_cell_editor
    exporting
      value = lr_input .

  call method lr_input->set_read_only
    exporting
      value = ABAP_TRUE
  .

when 'CURRENCY'.
  create object lr_input
    exporting
```

```
        value_fieldname = ls_name-name.
    call method lr_column->set_cell_editor
        exporting
            value = lr_input .
    call method lr_input->set_read_only
        exporting
            value = ABAP_TRUE .

when 'OTHERS'.
    create object lr_input
        exporting
            value_fieldname = ls_name-name.

    call method lr_column->set_cell_editor
        exporting
            value = lr_input .
    call method lr_input->set_read_only
        exporting
            value = ABAP_FALSE .

endcase.

* To make the required row is editable
    lr_input->set_read_only_fieldname( value = 'READ_ONLY' ).

endloop.

* Set the table Editable
lo_value->if_salv_wd_table_settings~set_read_only( value = abap_false ).

* delete column
call method lo_value->if_salv_wd_column_settings~delete_column
    exporting
        id      = 'READ_ONLY' .

Endmethod.
```

Output

Check the below output. The columns before the changed cell value and itself are not editable except the changed value in cell row and columns after the changed cell values are editable.

View [Standard View] Print Version Export Check Append Row Insert Row Delete Row											
ID	No.	Date	Airfare	Curr.	Plane	Capacity in economy class	Occupied economy class	Booking total	Capacity business class	Occupied bus	
AA	0017	27.02.2008	422.94	USD	747-400	385	357	193,812.47	31		
AA	0017	26.03.2008	422.94	USD	747-400	385	354	191,579.41	31		
AA	0017	23.04.2008	422.94	USD	747-400	385	356	193,846.29	31		
AA	0017	21.05.2008	422.94	USD	747-400	385	353	191,025.21	31		
AA	0017	18.06.2008	422.94	USD	747-400	385	349	192,230.75	31		
AA	0017	16.07.2008	560.00	USD	747-400	385	355	192,061.46	31		
AA	0017	13.08.2008	422.94	USD	747-400	385	355	194,129.57	31		
AA	0017	10.09.2008	422.94	USD	747-400	385	356	193,888.50	31		
AA	0017	08.10.2008	422.94	USD	747-400	385	353	192,010.80	31		
AA	0017	05.11.2008	422.94	USD	747-400	385	189	108,361.60	31		
AA	0017	03.12.2008	422.94	USD	747-400	385	37	29,191.34	31		
AA	0017	31.12.2008	422.94	USD	747-400	385	115	68,123.03	31		
AA	0017	28.01.2009	422.94	USD	747-400	385	30	24,864.72	31		
AA	0017	25.02.2009	422.94	USD	747-400	385	37	29,060.21	31		
AA	0017	25.03.2009	422.94	USD	747-400	385	5	11,677.39	31		
AA	0064	29.02.2008	422.94	USD	A310-300	280	253	135,294.41	22		
AA	0064	28.03.2008	422.94	USD	A310-300	280	253	135,053.37	22		
AA	0064	25.04.2008	422.94	USD	A310-300	280	253	134,380.84	22		
AA	0064	23.05.2008	422.94	USD	A310-300	280	247	131,166.55	22		
AA	0064	20.06.2008	422.94	USD	A310-300	280	247	130,299.55	22		
AA	0064	18.07.2008	422.94	USD	A310-300	280	253	134,169.42	22		
AA	0064	15.08.2008	422.94	USD	A310-300	280	253	133,539.19	22		
AA	0064	12.09.2008	422.94	USD	A310-300	280	248	130,430.56	22		
AA	0064	10.10.2008	422.94	USD	A310-300	280	253	133,814.07	22		

Related Content

[SAP List Viewer in Web Dynpro for ABAP](#)

For more information, visit the [User Interface Technology homepage](#).

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.