

## ALV Object Model – Simple 2D Table – Event Handling

### Applies to:

Netweaver 2004 and Netweaver 2004s

### Summary

This tutorial will show how to implement event handling when using the new ALV object model. For more examples, see any program which begins with SALV\* in your Netweaver ABAP System.

**Author(s):** Rich Heilman

**Company:** Yorktowne Cabinetry

**Created on:** 28 September 2006



Rich Heilman is an ABAP/J2EE Software Engineer/Analyst for Yorktowne Cabinetry, Inc. based in Red Lion, Pennsylvania, USA. He has a total of nine years experience in the IT industry. He has spent the past five years studying ABAP and Java.

## Table of Contents

Applies to: .....	1
Summary.....	1
The Basic Program .....	3
Set the Gui Status.....	4
Event Handlers - Event ADDED_FUNCTION .....	5
Event Handlers - Event DOUBLE_CLICK .....	8
Related Content.....	11
Disclaimer and Liability Notice.....	12

## The Basic Program

Starting with the program below, we will add coding to handle some events for the ALV Grid. In this example the events `DOUBLE_CLICK` and `ADDED_FUNCTION` will be handled.

```
report zalvom_demo3.

data: ispfli type table of spfli.
data: xspfli type spfli.

data: gr_table type ref to cl_salv_table.
data: gr_selections type ref to cl_salv_selections.

start-of-selection.

    select * into corresponding fields of table ispfli from spfli
           up to 100 rows.

    call method cl_salv_table=>factory
      importing
        r_salv_table = gr_table
      changing
        t_table      = ispfli.

* Set up selections.
gr_selections = gr_table->get_selections( ).
gr_selections->set_selection_mode( 1 ).    "Single

* Display
gr_table->display( ).
```

## Set the Gui Status

Next, go to function group SALV\_METADATA\_STATUS and copy the gui status SALV\_TABLE\_STANDARD into the ZALVOM\_DEMO3 program. This is the standard gui status for the 2 Dimensional Table ALV grid. Once you have copied the status, set the screen status using the appropriate method of the object GR\_TABLE. Go to the gui status and add a new button on the application toolbar and name it as "MYFUNCTION".

```
report zalvom_demo3.
```

```
data: ispfli type table of spfli.
```

```
data: xspfli type spfli.
```

```
data: gr_table type ref to cl_salv_table.
```

```
data: gr_selections type ref to cl_salv_selections.
```

```
start-of-selection.
```

```
select * into corresponding fields of table ispfli from spfli
       up to 100 rows.
```

```
call method cl_salv_table=>factory
```

```
  importing
```

```
    r_salv_table = gr_table
```

```
  changing
```

```
    t_table      = ispfli.
```

```
gr_table->set_screen_status(
  pfstatus      = 'SALV_TABLE_STANDARD'
  report        = sy-repid
  set_functions = gr_table->c_functions_all ).
```

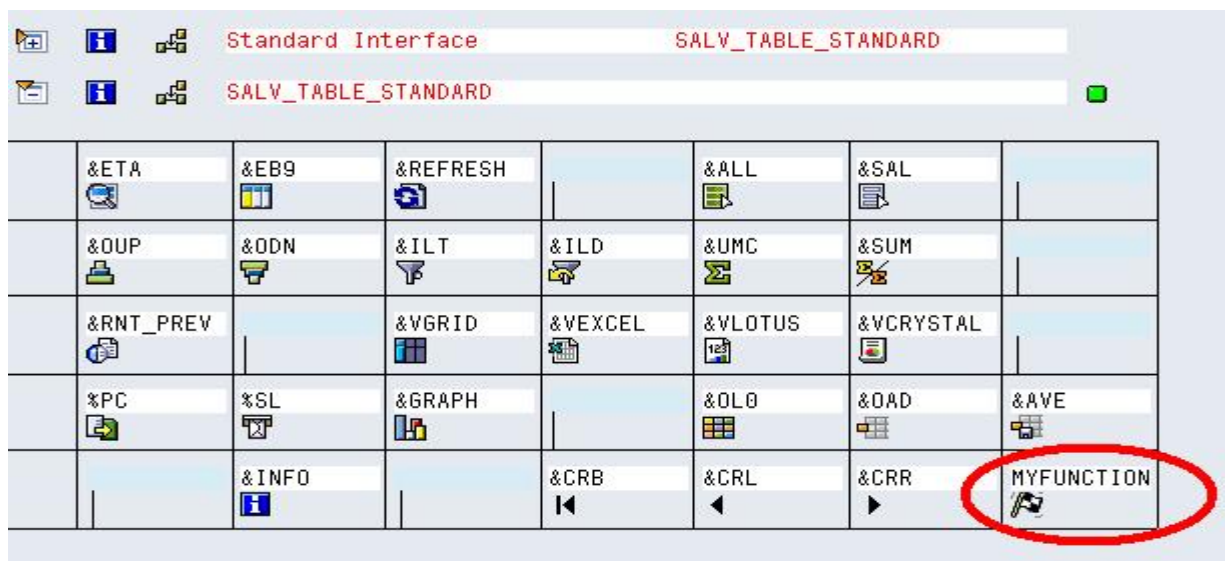
```
* Set up selections.
```

```
gr_selections = gr_table->get_selections( ).
```

```
gr_selections->set_selection_mode( 1 ).  "Single
```

```
* Display
```

```
gr_table->display( ).
```



## Event Handlers - Event ADDED\_FUNCTION

Next, create a local class which will act as the event handler, define the event handler method for the ADDED\_FUNCTION event. Define an object reference variable for the local class. Retrieve the events object from the GR\_TABLE, create the event handler object and set the handler method for the event. Finally, add the implementation for the ON\_USER\_COMMAND event handler method.

```
report zalvom_demo3.

data: ispfli type table of spfli.
data: xspfli type spfli.

data: gr_table type ref to cl_salv_table.
data: gr_events type ref to cl_salv_events_table.
data: gr_selections type ref to cl_salv_selections.

*-----*
*      CLASS lcl_handle_events DEFINITION
*-----*
*
*-----*
class lcl_handle_events definition.
  public section.
  methods:
    on_user_command for event added_function of cl_salv_events
      importing e_salv_function,

endclass.                "lcl_handle_events DEFINITION

data: event_handler type ref to lcl_handle_events.

start-of-selection.

  select * into corresponding fields of table ispfli from spfli
         up to 100 rows.

  call method cl_salv_table=>factory
    importing
      r_salv_table = gr_table
    changing
      t_table      = ispfli.

  gr_table->set_screen_status(
    pfstatus      = 'SALV_TABLE_STANDARD'
    report        = sy-repid
    set_functions = gr_table->c_functions_all ).

  gr_events = gr_table->get_event( ).

  create object event_handler.

  set handler event_handler->on_user_command for gr_events.

* Set up selections.
```

```

gr_selections = gr_table->get_selections( ).
gr_selections->set_selection_mode( 1 ).    "Single

* Display
gr_table->display( ).

*-----*
*      CLASS lcl_handle_events IMPLEMENTATION
*-----*
*
*-----*
class lcl_handle_events implementation.
method on_user_command.
* Get the selection rows
  data: lr_selections type ref to cl_salv_selections.
  data: lt_rows      type salv_t_row.
  data: ls_rows      type i.
  data: message type string.

  case e_salv_function.
    when 'MYFUNCTION'.

      lr_selections = gr_table->get_selections( ).
      lt_rows = lr_selections->get_selected_rows( ).

      read table lt_rows into ls_rows index 1.

      read table ispfli into xspfli index ls_rows.

      concatenate xspfli-carrid    xspfli-connid
                  xspfli-cityfrom xspfli-cityto
                  into message separated by space.

      message i001(00) with 'You pushed the button!' message.

    endcase.

  endmethod.                                "on_user_command
endclass.                                    "lcl_handle_events IMPLEMENTATION

```

Run the program, select a row by single clicking on it and click the icon for the new function that you added. Notice that some of the data in the row that was clicked is now showing in the message.

The screenshot shows the SAP ALV OM Demo 3 interface. The main table displays flight data with columns: Cli..., Air..., Flig..., Co..., Depart. city, De..., Co..., Arrival city, and Ta... The row for flight JL 407 is selected and highlighted in yellow. A red circle highlights this row. Below the table, an information message box is open, displaying the text: "You pushed the button [JL 0407 TOKYO FRANKFURT]". A red circle highlights the message content. The message box also contains a checkmark icon and a question mark icon.

Cli...	Air...	Flig...	Co...	Depart. city	De...	Co...	Arrival city	Ta...
000	AA	17	US	NEW YORK	JFK	US	SAN FRANCISCO	SFO
000	AA	64	US	SAN FRANCISCO	SFO	US	NEW YORK	JFK
000	AZ	555	IT	ROME	FCO	DE	FRANKFURT	FRA
000	AZ	788	IT	ROME	FCO	JP	TOKYO	TYO
000	AZ	789	JP	TOKYO	TYO	IT	ROME	FCO
000	AZ	790	IT	ROME	FCO	JP	OSAKA	KIX
000	DL	106	US	NEW YORK	JFK	DE	FRANKFURT	FRA
000	DL	1699	US	NEW YORK	JFK	US	SAN FRANCISCO	SFO
000	DL	1384	US	SAN FRANCISCO	SFO	US	NEW YORK	JFK
000	JL	407	JP	TOKYO	NRT	DE	FRANKFURT	FRA
000	JL	408	DE	FRANKFURT	FRA	JP	TOKYO	NRT
000	LH							
000	LH							
000	LH							
000	LH							
000	LH							
000	LH							
000	QF							
000	QF							
000	SG	2	SG	SINGAPORE	SIN	US	SAN FRANCISCO	SFO

## Event Handlers - Event DOUBLE\_CLICK

Define the event handler method for DOUBLE\_CLICK event and add the implementation for the ON\_DOUBLE\_CLICK event handler method. Remember to set the handler for the event.

```

report   zalvom_demo3.

data:   ispfli type table of spfli.
data:   xspfli type spfli.

data:   gr_table type ref to cl_salv_table.
data:   gr_functions type ref to cl_salv_functions_list.
data:   gr_events type ref to cl_salv_events_table.
data:   gr_selections type ref to cl_salv_selections.

*-----*
*       CLASS lcl_handle_events DEFINITION
*-----*
*
*-----*
class lcl_handle_events definition.
  public section.
  methods:
    on_user_command for event added_function of cl_salv_events
      importing e_salv_function,

      on_double_click for event double_click of cl_salv_events_table
        importing row column.

endclass.                                "lcl_handle_events DEFINITION

data:   event_handler type ref to lcl_handle_events.

start-of-selection.

  select * into corresponding fields of table ispfli from spfli
         up to 100 rows.

  call method cl_salv_table=>factory
    importing
      r_salv_table = gr_table
    changing
      t_table      = ispfli.

  gr_table->set_screen_status(
    pfstatus      = 'SALV_TABLE_STANDARD'
    report        = sy-repid
    set_functions = gr_table->c_functions_all ).

  gr_events = gr_table->get_event( ).

  create object event_handler.

  set handler event_handler->on_user_command for gr_events.

```



```

set handler event_handler->on_double_click for gr_events.

* Set up selections.
gr_selections = gr_table->get_selections( ).
gr_selections->set_selection_mode( 1 ).    "Single"

* Display
gr_table->display( ).

*-----*
*      CLASS lcl_handle_events IMPLEMENTATION
*-----*
*
*-----*
class lcl_handle_events implementation.
  method on_user_command.

* Get the selection rows
  data: lr_selections type ref to cl_salv_selections.
  data: lt_rows    type salv_t_row.
  data: ls_rows   type i.
  data: message type string.

  case e_salv_function.
    when 'MYFUNCTION'.

      lr_selections = gr_table->get_selections( ).
      lt_rows = lr_selections->get_selected_rows( ).

      read table lt_rows into ls_rows index 1.

      read table ispfli into xspfli index ls_rows.

      concatenate xspfli-carrid    xspfli-connid
                  xspfli-cityfrom xspfli-cityto
                  into message separated by space.

      message i001(00) with 'You pushed the button!' message.

    endcase.

  endmethod.                                "on_user_command"

  method on_double_click.

    data: message type string.
    data: row_c(4) type c.

    row_c = row.

    concatenate 'Row' row_c 'Column' column
                  into message separated by space.
    message i001(00) with 'You double-clicked on ' message.

  endmethod.                                "on_double_click"
endclass.                                  "lcl_handle_events IMPLEMENTATION"

```

Run the program, double click on the fifth row in the Depart. City column, notice the information message contains the row number and column name of the cell which you double clicked.

The screenshot shows the SAP ALV Object Model interface. The main window title is "ALV OM Demo 3". The table displays flight data with columns: Cli..., Air..., Flig..., Co..., Depart. city, De..., Co..., Arrival city, and Ta... The fifth row (000 AZ 789 JP TOKYO) is highlighted in yellow. The "Depart. city" column is circled in red. An information message box is open, displaying the text "You double-clicked on Row 5 Column CITYFROM:". The message box also contains a question mark icon and a close button.

Cli...	Air...	Flig...	Co...	Depart. city	De...	Co...	Arrival city	Ta...
000	AA	17	US	NEW YORK	JFK	US	SAN FRANCISCO	SFO
000	AA	64	US	SAN FRANCISCO	SFO	US	NEW YORK	JFK
000	AZ	555	IT	ROME	FCO	DE	FRANKFURT	FRA
000	AZ	788	IT	ROME	FCO	JP	TOKYO	TYO
000	AZ	789	JP	TOKYO	TYO	IT	ROME	FCO
000	AZ	790	IT	ROME	FCO	JP	OSAKA	KIX
000	D							RA
000	D							SFO
000	D							FK
000	JL							RA
000	JL							JRT
000	LH							FK
000	LH							RA
000	LH	402	DE	FRANKFURT	FRA	US	NEW YORK	JFK

## Related Content

[Help - ALV Object Model](#)

[Utilizing the New ALV Object Model](#)

[SDN ABAP Forum](#)

## 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.