

ALV Object Model – Simple 2D Table - The Basics

Applies to:

Netweaver 2004 and Netweaver 2004s

Summary

This tutorial is the first of the series, which deals with the ALV Object Model. In this tutorial, you will learn how to create a simple two-dimensional table report using the ALV Object Model.

Author(s): Rich Heilman

Company: Yorktowne Cabinetry

Created on: 21 September 2006

Author Bio



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
Author Bio	1
Main Class – CL_SALV_TABLE.....	3
Functions – CL_SALV_FUNCTIONS	4
Display Settings – CL_SALV_DISPLAY_SETTINGS.....	4
Columns – CL_SALV_COLUMNS_TABLE and CL_SALV_COLUMN_TABLE	5
Sorts – CL_SALV_SORTS	8
Aggregations – CL_SALV_AGGREGATIONS	10
Filters – CL_SALV_FILTERS	12
Layouts – CL_SALV_LAYOUT	14
Related Content.....	15
Disclaimer and Liability Notice.....	16

Main Class – CL_SALV_TABLE

The main class used to create the simple 2D table is the class CL_SALV_TABLE. Create a reference variable for this class. Create an internal table and fill this internal table with data as show below.

```
REPORT ZALVOM_DEM01.
```

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

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

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

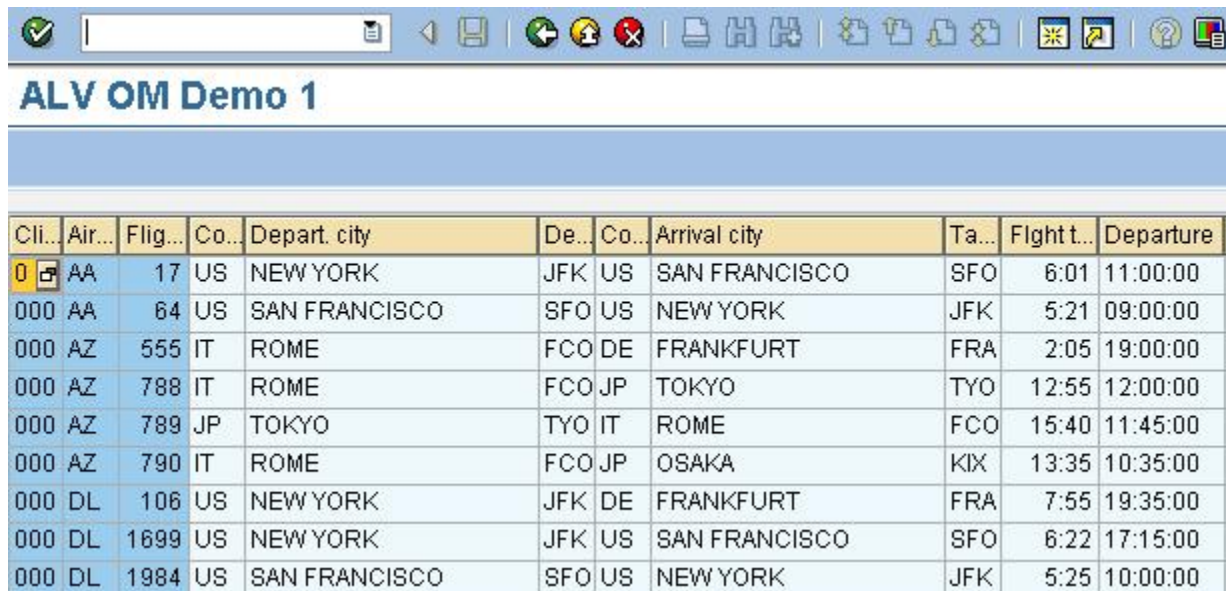
```
    select * into table ispfli from spfli.
```

Next we need to create the ALV object for the 2D table. The FACTORY method allows you to create the ALV object in 3 ways. You can create the ALV Grid, as a classical list display, as a full screen grid, and finally embedded into a screen container. For this example, we will be working with the full screen grid. Create the call to the FACTORY method. We are importing the object reference into GR_TABLE and passing the internal table ISPFLI.

```
    cl_salv_table=>factory( importing r_salv_table = gr_table
                           changing t_table      = ispfli ).
```

Next we need to display the grid, for this we use the DISPLAY method . Simply call it.

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



The screenshot shows the SAP ALV OM Demo 1 application. The title bar reads "ALV OM Demo 1". Below the title bar is a table displaying flight data. The table has 11 columns: Cli..., Air..., Flig..., Co..., Depart. city, De..., Co..., Arrival city, Ta..., Flight t..., and Departure. The data is as follows:

Cli...	Air...	Flig...	Co...	Depart. city	De...	Co...	Arrival city	Ta...	Flight t...	Departure
0	AA	17	US	NEW YORK	JFK	US	SAN FRANCISCO	SFO	6:01	11:00:00
000	AA	64	US	SAN FRANCISCO	SFO	US	NEW YORK	JFK	5:21	09:00:00
000	AZ	555	IT	ROME	FCO	DE	FRANKFURT	FRA	2:05	19:00:00
000	AZ	788	IT	ROME	FCO	JP	TOKYO	TYO	12:55	12:00:00
000	AZ	789	JP	TOKYO	TYO	IT	ROME	FCO	15:40	11:45:00
000	AZ	790	IT	ROME	FCO	JP	OSAKA	KIX	13:35	10:35:00
000	DL	106	US	NEW YORK	JFK	DE	FRANKFURT	FRA	7:55	19:35:00
000	DL	1699	US	NEW YORK	JFK	US	SAN FRANCISCO	SFO	6:22	17:15:00
000	DL	1984	US	SAN FRANCISCO	SFO	US	NEW YORK	JFK	5:25	10:00:00

Functions – CL_SALV_FUNCTIONS

Next, add functions to the application toolbar. For this, use the CL_SALV_FUNCTIONS class. Create the object reference variable and receive the object using the GET_FUNCTIONS method of the GR_TABLE object. Call the method SET_ALL to force the ALV grid to show all standard functions.

```
report zalvom_demo1.

data: ispfli type table of spfli.

data: gr_table      type ref to cl_salv_table.
data: gr_functions  type ref to cl_salv_functions.

start-of-selection.

    select * into table ispfli from spfli.

    cl_salv_table=>factory( importing r_salv_table = gr_table
                           changing t_table      = ispfli ).

    gr_functions = gr_table->get_functions( ).
    gr_functions->set_all( abap_true ).

    gr_table->display( ).
```

The result is now you have the standard buttons on the application toolbar.



Cli...	Air...	Flig...	Co...	Depart. city	De...	Co...	Arrival city	Ta...	Flight t...	Departure
000	AA	17	US	NEW YORK	JFK	US	SAN FRANCISCO	SFO	6:01	11:00:00
000	AA	64	US	SAN FRANCISCO	SFO	US	NEW YORK	JFK	5:21	09:00:00
000	AZ	555	IT	ROME	FCO	DE	FRANKFURT	FRA	2:05	19:00:00
000	AZ	788	IT	ROME	FCO	JP	TOKYO	TYO	12:55	12:00:00
000	AZ	789	JP	TOKYO	TYO	IT	ROME	FCO	15:40	11:45:00
000	AZ	790	IT	ROME	FCO	JP	OSAKA	KIX	13:35	10:35:00
000	DL	106	US	NEW YORK	JFK	DE	FRANKFURT	FRA	7:55	19:35:00
000	DL	1699	US	NEW YORK	JFK	US	SAN FRANCISCO	SFO	6:22	17:15:00
000	DL	1984	US	SAN FRANCISCO	SFO	US	NEW YORK	JFK	5:25	10:00:00

Display Settings – CL_SALV_DISPLAY_SETTINGS

Next, we can change some display settings using the class CL_SALV_DISPLAY_SETTINGS. Create the object reference variable and receive the object using the GET_DISPLAY_SETTINGS method of the GR_TABLE object. In this example, we are setting the "Striped Pattern" for the ALV Grid rows, and setting the heading in the title bar.

```
report zalvom_demo1.
```

```

data: ispfli type table of spfli.

data: gr_table      type ref to cl_salv_table.
data: gr_functions  type ref to cl_salv_functions.
data: gr_display    type ref to cl_salv_display_settings.

start-of-selection.

    select * into table ispfli from spfli.

    cl_salv_table=>factory( importing r_salv_table = gr_table
                          changing t_table      = ispfli ).

    gr_functions = gr_table->get_functions( ).
    gr_functions->set_all( abap_true ).

    gr_display = gr_table->get_display_settings( ).
    gr_display->set_striped_pattern( cl_salv_display_settings=>true ).
    gr_display->set_list_header( 'This is the heading' ).

    gr_table->display( ).

```



Cli...	Air...	Flig...	Co...	Depart. city	De...	Co...	Arrival city	Ta...	Flightt...	Departure
000	AA	17	US	NEW YORK	JFK	US	SAN FRANCISCO	SFO	6:01	11:00:00
000	AA	64	US	SAN FRANCISCO	SFO	US	NEW YORK	JFK	5:21	09:00:00
000	AZ	555	IT	ROME	FCO	DE	FRANKFURT	FRA	2:05	19:00:00
000	AZ	788	IT	ROME	FCO	JP	TOKYO	TYO	12:55	12:00:00
000	AZ	789	JP	TOKYO	TYO	IT	ROME	FCO	15:40	11:45:00
000	AZ	790	IT	ROME	FCO	JP	OSAKA	KIX	13:35	10:35:00
000	DL	106	US	NEW YORK	JFK	DE	FRANKFURT	FRA	7:55	19:35:00
000	DL	1699	US	NEW YORK	JFK	US	SAN FRANCISCO	SFO	6:22	17:15:00
000	DL	1984	US	SAN FRANCISCO	SFO	US	NEW YORK	JFK	5:25	10:00:00

Columns – CL_SALV_COLUMNS_TABLE and CL_SALV_COLUMN_TABLE

Next, we can change some of the attributes of a specific column in the ALV grid. In this example we will change the Heading Text of a column as well as the color of a column. Create the object reference variable and receive the object using the GET_COLUMNS method of the GR_TABLE object. This will pass you the object for all columns of the ALV grid. To access just one column, call the method GET_COLUMN from the GR_COLUMNS object. In this example, we are accessing the CITYTO column and the CITYFROM column.

```

report zalvom_demo1.

data: ispfli type table of spfli.

data: gr_table      type ref to cl_salv_table.
data: gr_functions  type ref to cl_salv_functions.

```

```
data: gr_display    type ref to cl_salv_display_settings.
data: gr_columns    type ref to cl_salv_columns_table.
data: gr_column     type ref to cl_salv_column_table.
```

```
data: color type lvc_s_colo.
```

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

```
select * into table ispfli from spfli.
```

```
cl_salv_table=>factory( importing r_salv_table = gr_table
                        changing t_table      = ispfli ).
```

```
gr_functions = gr_table->get_functions( ).
gr_functions->set_all( abap_true ).
```

```
gr_display = gr_table->get_display_settings( ).
gr_display->set_striped_pattern( cl_salv_display_settings=>true ).
gr_display->set_list_header( 'This is the heading' ).
```

```
gr_columns = gr_table->get_columns( ).
gr_column ?= gr_columns->get_column( 'CITYTO' ).
```

```
gr_column->set_long_text( 'This is long text' ).
gr_column->set_medium_text( 'This is med text' ).
gr_column->set_short_text( 'This is sh' ).
```

```
gr_column ?= gr_columns->get_column( 'CITYFROM' ).
color-col = '6'.
color-int = '1'.
color-inv = '0'.
gr_column->set_color( color ).
```

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




This is the heading



Cli...	Air...	Flig...	Co...	Depart. city	De...	Co...	This is med text	Ta...	Fightt...	Departure
0	AA	17	US	NEW YORK	JFK	US	SAN FRANCISCO	SFO	6:01	11:00:00
000	AA	64	US	SAN FRANCISCO	SFO	US	NEW YORK	JFK	5:21	09:00:00
000	AZ	555	IT	ROME	FCO	DE	FRANKFURT	FRA	2:05	19:00:00
000	AZ	788	IT	ROME	FCO	JP	TOKYO	TYO	12:55	12:00:00
000	AZ	789	JP	TOKYO	TYO	IT	ROME	FCO	15:40	11:45:00
000	AZ	790	IT	ROME	FCO	JP	OSAKA	KIX	13:35	10:35:00
000	DL	106	US	NEW YORK	JFK	DE	FRANKFURT	FRA	7:55	19:35:00
000	DL	1699	US	NEW YORK	JFK	US	SAN FRANCISCO	SFO	6:22	17:15:00
000	DL	1984	US	SAN FRANCISCO	SFO	US	NEW YORK	JFK	5:25	10:00:00

Sorts – CL_SALV_SORTS

Next, we can add some sorting to the ALV grid. Create the object reference variable and receive the object using the GET_SORTS method of the GR_TABLE object. Next, add the sort by calling the ADD_SORT method of the GR_SORTS object.

```
report zalvom_demo1.

data: ispfli type table of spfli.

data: gr_table      type ref to cl_salv_table.
data: gr_functions  type ref to cl_salv_functions.
data: gr_display    type ref to cl_salv_display_settings.
data: gr_columns    type ref to cl_salv_columns_table.
data: gr_column     type ref to cl_salv_column_table.
data: gr_sorts      type ref to cl_salv_sorts.

data: color type lvc_s_color.

start-of-selection.

  select * into table ispfli from spfli.

  cl_salv_table=>factory( importing r_salv_table = gr_table
                        changing t_table      = ispfli ).

  gr_functions = gr_table->get_functions( ).
  gr_functions->set_all( abap_true ).

  gr_display = gr_table->get_display_settings( ).
  gr_display->set_striped_pattern( cl_salv_display_settings=>true ).
  gr_display->set_list_header( 'This is the heading' ).

  gr_columns = gr_table->get_columns( ).
  gr_column ?= gr_columns->get_column( 'CITYTO' ).

  gr_column->set_long_text( 'This is long text' ).
  gr_column->set_medium_text( 'This is med text' ).
  gr_column->set_short_text( 'This is sh' ).

  gr_column ?= gr_columns->get_column( 'CITYFROM' ).
  color-col = '6'.
  color-int = '1'.
  color-inv = '0'.
  gr_column->set_color( color ).

  gr_sorts = gr_table->get_sorts( ).
  gr_sorts->add_sort 'CITYTO' ).

  gr_table->display( ).
```


✓ | |

This is the heading

Cli...	Air...	Flig...	Co...	Depart. city	De...	Co...	This is med text	Ta...	Flight t...	Departure
0	LH	2402	DE	FRANKFURT	FRA	DE	BERLIN	SXF	1:05	10:30:00
000	AZ	555	IT	ROME	FCO	DE	FRANKFURT	FRA	2:05	19:00:00
000	DL	106	US	NEW YORK	JFK	DE		FRA	7:55	19:35:00
000	JL	407	JP	TOKYO	NRT	DE		FRA	12:05	13:30:00
000	LH	401	US	NEW YORK	JFK	DE		FRA	7:15	18:30:00
000	LH	2407	DE	BERLIN	TXL	DE		FRA	1:05	07:10:00
000	QF	5	SG	SINGAPORE	SIN	DE		FRA	13:45	22:50:00
000	UA	3504	US	SAN FRANCISCO	SFO	DE		FRA	10:30	15:00:00
000	UA	3516	US	NEW YORK	JFK	DE		FRA	7:25	16:20:00
000	SQ	158	SG	SINGAPORE	SIN	ID	JAKARTA	JKT	1:35	15:25:00

Aggregations – CL_SALV_AGGREGATIONS

Since we sorted by CITYTO, we can add an aggregation to subtotal the DISTANCE by CITYTO. Create the object reference variable and receive the object using the GET_AGGREGATIONS method of the GR_TABLE object. Next, add the aggregation by calling the ADD_AGGREGATION method of the GR_SORTS object. We also need to modify the call to ADD_SORT to set the SUBTOTAL = ABAP_TRUE.

```
report  zalvom_demo1.

data: ispfli type table of spfli.

data: gr_table      type ref to cl_salv_table.
data: gr_functions  type ref to cl_salv_functions.
data: gr_display    type ref to cl_salv_display_settings.
data: gr_columns    type ref to cl_salv_columns_table.
data: gr_column     type ref to cl_salv_column_table.
data: gr_sorts      type ref to cl_salv_sorts.
data: gr_agg        type ref to cl_salv_aggregations.

data: color type lvc_s_color.

start-of-selection.

  select * into table ispfli from spfli.

  cl_salv_table=>factory( importing r_salv_table = gr_table
                        changing t_table      = ispfli ).

  gr_functions = gr_table->get_functions( ).
  gr_functions->set_all( abap_true ).

  gr_display = gr_table->get_display_settings( ).
  gr_display->set_stripped_pattern( cl_salv_display_settings=>true ).
  gr_display->set_list_header( 'This is the heading' ).

  gr_columns = gr_table->get_columns( ).
  gr_column ?= gr_columns->get_column( 'CITYTO' ).

  gr_column->set_long_text( 'This is long text' ).
  gr_column->set_medium_text( 'This is med text' ).
  gr_column->set_short_text( 'This is sh' ).

  gr_column ?= gr_columns->get_column( 'CITYFROM' ).
  color-col = '6'.
  color-int = '1'.
  color-inv = '0'.
  gr_column->set_color( color ).

  gr_sorts = gr_table->get_sorts( ).
  gr_sorts->add_sort( columnname = 'CITYTO' subtotal = abap_true ).

  gr_agg = gr_table->get_aggregations( ).
  gr_agg->add_aggregation( 'DISTANCE' ).
```

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

Cli...	Air...	Flig...	Co...	Depart. city	De...	Co...	This is med text	Ta...	Distance	Dis.
000	LH	2402	DE	FRANKFURT	FRA	DE	BERLIN	SXF	555	KM
							BERLIN		555	KM
000	AZ	555	IT	ROME	FCO	DE	FRANKFURT	FRA	845	MI
000	DL	106	US	NEW YORK	JFK	DE		FRA	3,851	MI
000	JL	407	JP	TOKYO	NRT	DE		FRA	9,100	KM
000	LH	401	US	NEW YORK	JFK	DE		FRA	6,162	KM
000	LH	2407	DE	BERLIN	TXL	DE		FRA	555	KM
000	QF	5	SG	SINGAPORE	SIN	DE		FRA	10,000	KM
000	UA	3504	US	SAN FRANCISCO	SFO	DE		FRA	5,685	MI
000	UA	3516	US	NEW YORK	JFK	DE		FRA	6,162	KM
							FRANKFURT		31,979	KM
									10,381	MI

Filters – CL_SALV_FILTERS

Using the CL_SALV_FILTERS class we can setup some filters for the data in our ALV GRID. Create the object reference variable and receive the object using the GET_FILTERS method of the GR_TABLE object, and then simply called the method ADD_FILTER with the parameters.

```
report zalvom_demo1.

data: ispfli type table of spfli.

data: gr_table      type ref to cl_salv_table.
data: gr_functions  type ref to cl_salv_functions.
data: gr_display    type ref to cl_salv_display_settings.
data: gr_columns    type ref to cl_salv_columns_table.
data: gr_column     type ref to cl_salv_column_table.
data: gr_sorts      type ref to cl_salv_sorts.
data: gr_agg        type ref to cl_salv_aggregations.
data: gr_filter     type ref to cl_salv_filters.

data: color type lvc_s_colo.

start-of-selection.

  select * into table ispfli from spfli.

  cl_salv_table=>factory( importing r_salv_table = gr_table
                        changing t_table      = ispfli ).

  gr_functions = gr_table->get_functions( ).
  gr_functions->set_all( abap_true ).

  gr_display = gr_table->get_display_settings( ).
  gr_display->set_stripped_pattern( cl_salv_display_settings=>true ).
  gr_display->set_list_header( 'This is the heading' ).

  gr_columns = gr_table->get_columns( ).
  gr_column ?= gr_columns->get_column( 'CITYTO' ).

  gr_column->set_long_text( 'This is long text' ).
  gr_column->set_medium_text( 'This is med text' ).
  gr_column->set_short_text( 'This is sh' ).

  gr_column ?= gr_columns->get_column( 'CITYFROM' ).
  color-col = '6'.
  color-int = '1'.
  color-inv = '0'.
  gr_column->set_color( color ).

  gr_sorts = gr_table->get_sorts( ).
  gr_sorts->add_sort( columnname = 'CITYTO' subtotal = abap_true ).


  gr_agg = gr_table->get_aggregations( ).
  gr_agg->add_aggregation( 'DISTANCE' ).
```

```


gr_filter = gr_table->get_filters( ).
gr_filter->add_filter( columnname = 'CARRID' low = 'LH' ).




gr_table->display( ).


```



This is the heading



Cli...	A...	Flig...	Co...	Depart. city	De...	Co...	This is med text	Ta...	s	Distance	Dis.
000	LH	2402	DE	FRANKFURT	FRA	DE	BERLIN	SXF		555	KM
							BERLIN		■	555 KM	
000	LH	401	US	NEW YORK	JFK	DE	FRANKFURT	FRA		6,162	KM
000	LH	2407	DE	BERLIN	TXL	DE		FRA		555	KM
							FRANKFURT		■	6,717 KM	
000	LH	400	DE	FRANKFURT	FRA	US	NEW YORK	JFK		6,162	KM
000	LH	402	DE	FRANKFURT	FRA	US		JFK		6,162	KM
							NEW YORK		■	12,324 KM	
									■ ■	19,596 KM	



Layouts – CL_SALV_LAYOUT

If you want to allow the user to manage layouts of the ALV grid, you must use the class CL_SALV_LAYOUT. Create the object reference variable and receive the object using the GET_LAYOUT method of the GR_TABLE object. Then simply call the method SET_KEY with the parameters and set the save restriction using the SET_SAVE_RESTRICTION method.

```
report   zalvom_demo1.

data: ispfli type table of spfli.

data: gr_table      type ref to cl_salv_table.
data: gr_functions  type ref to cl_salv_functions.
data: gr_display    type ref to cl_salv_display_settings.
data: gr_columns    type ref to cl_salv_columns_table.
data: gr_column     type ref to cl_salv_column_table.
data: gr_sorts      type ref to cl_salv_sorts.
data: gr_agg        type ref to cl_salv_aggregations.
data: gr_filter     type ref to cl_salv_filters.
data: gr_layout     type ref to cl_salv_layout.

data: color type lvc_s_color.
data: key type salv_s_layout_key.

start-of-selection.

  select * into table ispfli from spfli.

  cl_salv_table=>factory( importing r_salv_table = gr_table
                        changing t_table      = ispfli ).

  gr_functions = gr_table->get_functions( ).
  gr_functions->set_all( abap_true ).

  gr_display = gr_table->get_display_settings( ).
  gr_display->set_stripped_pattern( cl_salv_display_settings=>true ).
  gr_display->set_list_header( 'This is the heading' ).

  gr_columns = gr_table->get_columns( ).
  gr_column ?= gr_columns->get_column( 'CITYTO' ).

  gr_column->set_long_text( 'This is long text' ).
  gr_column->set_medium_text( 'This is med text' ).
  gr_column->set_short_text( 'This is sh' ).

  gr_column ?= gr_columns->get_column( 'CITYFROM' ).
  color-col = '6'.
  color-int = '1'.
  color-inv = '0'.
  gr_column->set_color( color ).

  gr_sorts = gr_table->get_sorts( ).
  gr_sorts->add_sort( columnname = 'CITYTO' subtotal = abap_true ).

  gr_agg = gr_table->get_aggregations( ).
```



```

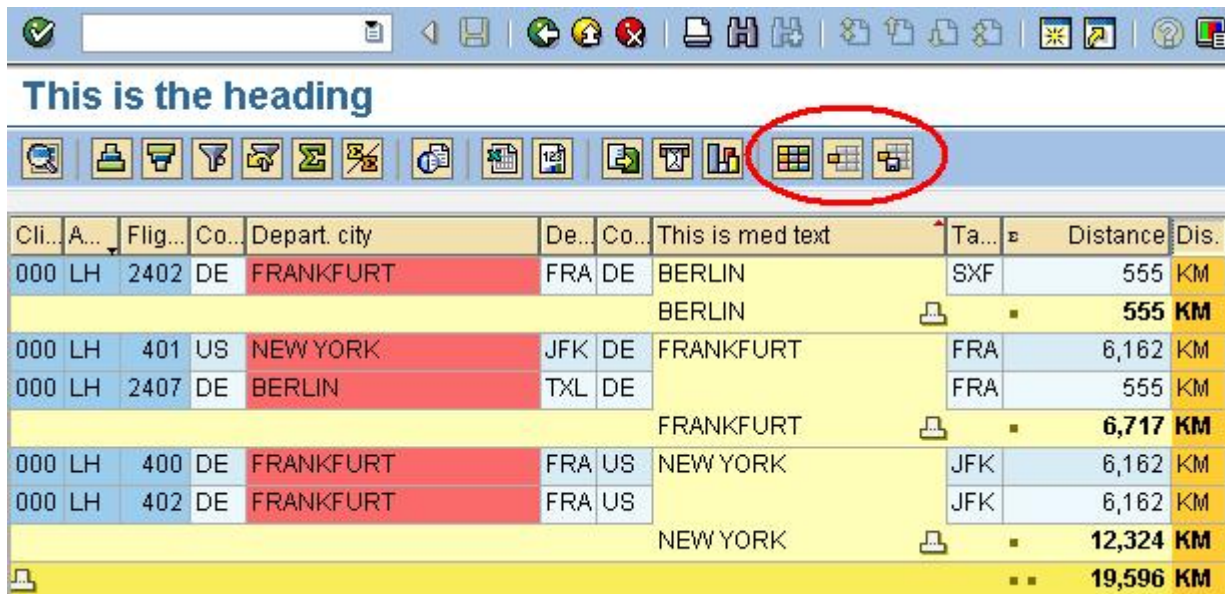
gr_agg->add_aggregation( 'DISTANCE' ).

gr_filter = gr_table->get_filters( ).
gr_filter->add_filter( columnname = 'CARRID' low = 'LH' ).

gr_layout = gr_table->get_layout( ).
key-report = sy-repid.
gr_layout->set_key( key ).
gr_layout->set_save_restriction( cl_salv_layout=>restrict_none ).

gr_table->display( ).

```



This is the heading

Cli...	A...	Flig...	Co...	Depart. city	De...	Co...	This is med text	Ta...	Distance	Dis.
000	LH	2402	DE	FRANKFURT	FRA	DE	BERLIN	SXF	555	KM
							BERLIN		555	KM
000	LH	401	US	NEW YORK	JFK	DE	FRANKFURT	FRA	6,162	KM
000	LH	2407	DE	BERLIN	TXL	DE	FRANKFURT	FRA	555	KM
							FRANKFURT		6,717	KM
000	LH	400	DE	FRANKFURT	FRA	US	NEW YORK	JFK	6,162	KM
000	LH	402	DE	FRANKFURT	FRA	US	NEW YORK	JFK	6,162	KM
							NEW YORK		12,324	KM
									19,596	KM

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.