




[Local Definitions/Implementations](#)
[Local Test Classes](#)
[Class documentation](#)
[Text elements](#)

Properties Interfaces Friends Attributes **Methods** Events Types Aliases

Type	Lo...	Line	Description
		84	Program ZPRG_FACTORY The exception CX_SALV_MSG is not caught.



## Class Builder: Display Class CL\_SALV\_TABLE

Local Definitions/Implementations Local Test Classes Class documentation Text elements

Class/Interface CL\_SALV\_TABLE Implemented / Active

Properties Interfaces Friends Attributes Methods Events Types Aliases

Parameters of Method FACTORY

Parameter	Type	P...	O...	Typing Method	Associated Type	Default Value	Description
LIST_DISPLAY	Importing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	SAP_BOOL	IF_SALV_C_BOOL_SAP=>FALSE	ALV Displayed in List Mode
R_CONTAINER	Importing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type Ref To	CL_GUI_CONTAINER		Abstract Container for GUI Controls
CONTAINER_NAME	Importing	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Type	STRING		
R_SALV_TABLE	Exporting	<input type="checkbox"/>	<input type="checkbox"/>	Type Ref To	CL_SALV_TABLE		Base Class Simple ALV Tables
T_TABLE	Changing	<input type="checkbox"/>	<input type="checkbox"/>	Type	TABLE		
		<input type="checkbox"/>	<input type="checkbox"/>	Type			
		<input type="checkbox"/>	<input type="checkbox"/>	Type			
		<input type="checkbox"/>	<input type="checkbox"/>	Type			
		<input type="checkbox"/>	<input type="checkbox"/>	Type			
		<input type="checkbox"/>	<input type="checkbox"/>	Type			
		<input type="checkbox"/>	<input type="checkbox"/>	Type			
		<input type="checkbox"/>	<input type="checkbox"/>	Type			
		<input type="checkbox"/>	<input type="checkbox"/>	Type			



1 warning for Program ZPRG\_FACTORY

Type	Lo...	Line	Description
		84	Program ZPRG_FACTORY
			The exception CX_SALV_MSG is not caught.

MethodEditGotoUtilitiesEnvironmentSystemHelp

✓

<<

**Class Builder**

**Class CL\_SALV\_TABLE**

**Display**

PatternPretty PrinterSignaturePublic SectionProtected SectionPrivate SectionText Elements

Ty.	Parameter	Typing	Description
	LIST_DISPLAY	TYPE SAP_BOOL DEFAULT IF_SALV_C_BOOL_SAP=>FALSE	ALV Displayed in List Mode
	R_CONTAINER	TYPE REF TO CL_GUI_CONTAINER OPTIONAL	Abstract Container for GUI Controls
	CONTAINER_NAME	TYPE STRING OPTIONAL	
	R_SALV_TABLE	TYPE REF TO CL_SALV_TABLE	Basis Class Simple ALV Tables
	T_TABLE	TYPE TABLE	
	CX_SALV_MSG		ALV: General Error Class with Message

MethodFACTORYactive

```
1  METHOD factory .
2
3      IF list display EQ abap_true.
4          DATA(lv_display_type) = if_salv_c_table_objects=>list.
5      ELSEIF r_container IS SUPPLIED.
6          lv_display_type = if_salv_c_table_objects=>grid.
7      ELSE.
8          lv_display_type = if_salv_c_table_objects=>fullscreen_grid.
9      ENDIF.
10
11      cl_alv_customizing=>set_one_ux_appearance( abap_true ).
12
13      cl_salv_gui_om_table_info=>factory(
14          ev_dynpro_id
```

Scope: \METHOD factoryABAPLn 1 Col 1

1 warning for Program ZPRG\_FACTORY

Type	Lo.	Line	Description
		84	Program ZPRG_FACTORY The exception CX_SALV_MSG is not caught.

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

Clipboard: Cut, Copy, Paste, Format Painter

Font: Calibri, 16, Bold, Italic, Underline, Text Color, Background Color

Alignment: Wrap Text, Merge & Center

Number: General, Percentage, Decimals

Styles: Conditional Formatting, Format as Table, Cell Styles

Cells: Insert, Delete, Format

Editing: AutoSum, Fill, Clear, Sort & Filter, Find & Select, Add-ins

E4 CL\_SALV\_TABLE

	E	F	G	H	I
1		ALV Object Model			
2					
3					
4	CL_SALV_TABLE	CL_SALV_HIERSEQ_TABLE	CL_SALV_TREE		
5	REUSE_ALV_LIST_DISPLAY	REUSE_ALV_HIERSEQ_LIST_DISPLAY			
6	REUSE_ALV_GRID_DISPLAY				
7	CL_GUI_ALV_GRID				
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					

Sheet1



### 3) Event based Functionality in ALV OOPS

Double clicking, HotSpot clicking etc.

#### ALV by CL\_SALV\_TABLE Class

- 1) CL\_SALV\_TABLE class is a part of ALV object model.
- 2) ALV object model is an encapsulation of the the pre-existing ALV tools.
- 3) For Example - **CL\_SALV\_TABLE** actually combines both the CL\_GUI\_ALV\_GRID class for container implementation, as well as the REUSE\_ALV\_GRID\_DISPLAY and REUSE\_ALV\_LIST\_DISPLAY function modules for full screen display.

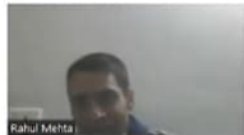
Advantage of ALV Object Model : It provides a collective interface to ALV tools.

Disadvange of ALV Object Model : We can not go for editable ALV using ALV object model.

#### Steps to Create ALV by CL\_SALV\_TABLE

Step1 : Call the FACTORY Method of CL\_SALV\_TABLE to get the instance of ALV table object.

Step2 : Call the DISPLAY Method of CL\_SALV\_TABLE to display the ALV.



- 1) In Abstract class, we have at least one abstract method, others can be non-abstract methods.
  - 2) Method visibility can be - Public, Private, Protected
  - 3) We need to redefine the abstract method in to sub classes to perform the implementation in Sub class.
- Important Point : We can not redefine a static method.
- 4) Multiple Inheritance is not possible.



#### Interface ->

- 1) All methods are abstract methods in Interface.
- 2) All methods are public.
- 3) There is no need to click on the redefine button( redefine button is disabled) , We can write the logic in class by double
- 4) Multiple Inheritance is possible.

11) Events -> By Which method of one class can call method of another class.

Triggering Method - Which will raise the event.

Event Handler Method -> Which will handle the event.

Register the event handler method using SET HANDLER statement.

Syntax for Event Handler.

SET HANDLER lo\_object1->event\_handler for lo\_object2.

Where lo\_object1 = Class Object in Which Event handler method is defined.

Event\_handler -> Event handler method name.

lo\_object2-> Class Object in which triggering method is defined.

#### 12) CONSTRUCTORS:



8 ABAP OOPS - Notepad

File Edit Format View Help

2) ALV object model is an encapsulation of the the pre-existing ALV tools.

3) For Example - CL\_SALV\_TABLE actually combines both the CL\_GUI\_ALV\_GRID class for container implementation, as well as the REUSE\_ALV\_GRID\_DISPLAY and REUSE\_ALV\_LIST\_DISPLAY function modules for full screen display.

Advantage of ALV Object Model : It provides a collective interface to ALV tools.

Disadvange of ALV Object Model : We can not go for editable ALV using ALV object model.

Steps to Create ALV by CL\_SALV\_TABLE

Step1 : Call the FACTORY Method of CL\_SALV\_TABLE to get the instance of ALV table object.

Step2 : Call the DISPLAY Method of CL\_SALV\_TABLE to display the ALV.

Important Links :

Link for Static and Instance Methods:

<http://www.sapspot.com/abap-objects-for-beginners-part-2-static-vs-instance-components/>

Link for Interfaces.

Rahul Mehta

Ln 242, Col 63

100%

Windows (CRLF)

UTF-8

21°C Haze

13:44

16-12-2023



## ABAP Editor: Change Report ZOOPS\_FACTORY\_CONTAINER

← → ↺ ↻ 📄 📁 📂 📅 📆 📇 📈 📉 📊 📋 📌 📍 📎 📏 📐 📑 📒 📓 📔 📕 📖 📗 📙 📚 📛 📜 📝 📞 📟 📠 📡 📢 📣 📤 📥 📦 📧 📨 📩 📪 📫 📬 📭 📮 📯 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📽 📾 📿 📠 📡 📢 📣 📤 📥 📦 📧 📨 📩 📪 📫 📬 📭 📮 📯 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📽 📾 📿

Report ZOOPS\_FACTORY\_CONTAINER Active

```

1  *&-----*
2  *& Report ZOOPS_FACTORY_CONTAINER
3  *&-----*
4  *&
5  *&-----*
6  REPORT zoops_factory_container.
7
8  *&-----*
9  *& Report ZPRG_FACTORY
10 *&-----*
11 *&
12 *&-----*
13 *REPORT zoops_factory_container.
14
15 TABLES: vbak, vbap.
16
17 TYPES: BEGIN OF lty_vbak,
18         vbeln TYPE vbeln_va,
19         erdat TYPE erdat,
20         erzet TYPE erzet,
21         ernam TYPE ernam,
22         vbtyp TYPE vbtyp1,
23         END OF lty_vbak.
24
25 DATA: lt_vbak TYPE TABLE OF lty_vbak,
26        ls_vbak TYPE lty_vbak.
27
28 TYPES: BEGIN OF lty_vbap,
29         vbeln TYPE vbeln_va,
30         posnr TYPE posnr_va,
31         matnr TYPE matnr,
32         END OF lty_vbap.
33
34 DATA: lt_vbap TYPE TABLE OF lty_vbap,
35        ls_vbap TYPE lty_vbap.
36
37 TYPES: BEGIN OF lty_final

```



```
35     ls_vbap TYPE lty_vbap.
36
37 □ TYPES: BEGIN OF lty_final,
38     vbeln TYPE vbeln_va,
39     erdat TYPE erdat,
40     erzet TYPE erzet,
41     ernam TYPE ernam,
42     vbtyp TYPE vbtyp1,
43     posnr TYPE posnr_va,
44     matnr TYPE matnr,
45     END OF lty_final.
46
47 DATA: lt_final TYPE TABLE OF lty_final,
48       ls_final TYPE lty_final.
49
50 DATA: lv_index TYPE n.
51
52 DATA: obj_factory TYPE REF TO cl_salv_table.
53
54 DATA: obj_container TYPE REF TO cl_gui_custom_container.
55
56 DATA: obj_function TYPE REF TO cl_salv_functions_list.
57
```

```
57
58  SELECT-OPTIONS:s_vbeln FOR vbak-vbeln.
59
60  START-OF-SELECTION.
61      SELECT vbeln erdat erzet ernam vbtyp
62          FROM vbak
63          INTO TABLE lt_vbak
64          WHERE vbeln IN s_vbeln.
65  IF lt_vbak IS NOT INITIAL.
66      SELECT vbeln posnr matnr
67          FROM vbap
68          INTO TABLE lt_vbap
69          FOR ALL ENTRIES IN lt_vbak
70          WHERE vbeln = lt_vbak-vbeln.
71
72  ENDIF.
```

```
72  ENDIF.  
73  SORT lt_vbak BY vbeln.  
74  SORT lt_vbap BY vbeln.  
75  LOOP AT lt_vbak INTO ls_vbak.  
76      READ TABLE lt_vbap INTO ls_vbap WITH KEY vbeln = ls_vbak-vbeln.  
77      IF sy-subrc = 0.  
78          lv_index = sy-tabix.  
79      ENDIF.  
80      LOOP AT lt_vbap INTO ls_vbap FROM lv_index.  
81          IF ls_vbak-vbeln <> ls_vbap-vbeln.  
82              EXIT.  
83          ELSE.  
84              ls_final-vbeln = ls_vbak-vbeln.  
85              ls_final-erdat = ls_vbak-erdat.  
86              ls_final-erzet = ls_vbak-erzet.  
87              ls_final-ernam = ls_vbak-ernam.  
88              ls_final-vbtyp = ls_vbak-vbtyp.  
89              ls_final-posnr = ls_vbap-posnr.  
90              ls_final-matnr = ls_vbap-matnr.  
91              APPEND ls_final TO lt_final.  
92              CLEAR:ls_final.  
93          ENDIF.  
94      ENDLOOP.  
95  ENDLOOP.  
96
```

```

96 *call screen '1010'.
97 □ TRY.
98     CALL METHOD cl_salv_table=>factory
99     EXPORTING
100         list_display = IF_SALV_C_BOOL_SAP=>FALSE
101         r_container   = obj_container
102         container_name = 'CON_NAM'
103     IMPORTING
104         r_salv_table = obj_factory
105     CHANGING
106         t_table      = lt_final.
107 □ CATCH cx_salv_msg.
108     ENDTRY.
109
110 * create OBJECT obj_value.
111 CALL METHOD obj_factory->if_salv_gui_om_table_info~get_functions
112 RECEIVING
113     value = obj_function.
114
115 CALL METHOD obj_function->set_default
116 EXPORTING
117     value = if_salv_c_bool_sap=>true. " Boolean Variable (X=True, Space=False)
118
119 CALL METHOD obj_factory->if_salv_gui_om_table_action~display
120 □ * IMPORTING
121     * exit_caused_by_caller =
122     * exit_caused_by_user   =
123     .
124
125
126 CALL SCREEN '1010'.
127
128 INCLUDE zoops_factory_container_stao01.
129
130 INCLUDE zoops_factory_container_usei01.
131

```

**BACK TO SELECTION SCREEN**

Sales Document	Created On	Time	Created By	Doc....	Item	Material
1	08.01.2025	08:08:19	BEST	C	10	00000000000000000026
2	08.01.2025	10:21:59	BEST	C	10	00000000000000000001
3	08.01.2025	12:04:26	BEST	C	10	00000000000000000001
4	09.01.2025	19:07:27	BEST	C	10	00000000000000000001
5	09.01.2025	23:51:47	BEST	C	10	00000000000000000062
6	10.01.2025	00:05:34	BEST	C	10	00000000000000000062
7	10.01.2025	00:17:25	BEST	C	10	00000000000000000075
8	10.01.2025	08:04:17	BEST	C	10	00000000000000000001
9	10.01.2025	08:07:38	BEST	C	10	00000000000000000001