Capgemini

ABAP Part II

Lesson 02: ABAP List Viewer

Lesson Objectives

After completing this lesson, participants will be able to understand the following -

- Control Framework
- ALV Grid
- Non Event Based Functionality
- Event Based Functionality

Introduction

The common features of report are column alignment, sorting, filtering, subtotals, totals etc.

To implement these, a lot of coding and logic is to be put.

To avoid that we can use a concept called ABAP List Viewer (ALV).

ALV (ABAP List Viewer) is a grid control, used for displaying lists.

The tool provides common list operations as generic functions and enhanced by user-defined options.

The grid control itself consists of a toolbar, a title and the output table displayed in a grid control. The user has control over the look of the grid (to certain degree) by the use of layout variants

Functions Provided by ALV Grid Control

Display non-hierarchical lists consistently with a modern design

Use typical list functions -such as sorting and filtering without extra programming effort

Adapt predefined list functions and their enhancements

Program responses to user actions (such as double-clicking a line) individually

Features

Navigating Within the List

Sorting in Ascending/Descending Order

Selecting and Deselecting Rows

Defining Exceptions

Setting and Deleting Filters

Displaying and Deleting Sums

Creating Subtotals

Optimizing the Column Width

Types of ALV Reports

Using ALV, we can have three types of reports:

- Simple Report
- Block Report
- Hierarchical Sequential Report

Simple Report

There are some function modules which will enable to produce the above reports without much effort.

All the definitions of internal tables, structures and constants are declared in a type-pool called SLIS.

The important function modules are:

- Reuse_alv_fieldcatalog_merge
- Reuse_alv_list_display
- Reuse_alv_grid_display

Simple Report

Reuse_ALV_fieldcatalog_merge:

- This function module is used to populate a fieldcatalog which is essential to display the data in ALV.
- If the output data is from a single dictionary table and all the columns are selected, then we need not exclusively create the field catalog.
- Its enough to mention the table name as a parameter(I_structure_name) in the REUSE_ALV_LIST_DISPLAY.
- In other cases, it has to be created.

Simple Report

REUSE_ALV_LIST_DISPLAY: This is the function module which prints the data.

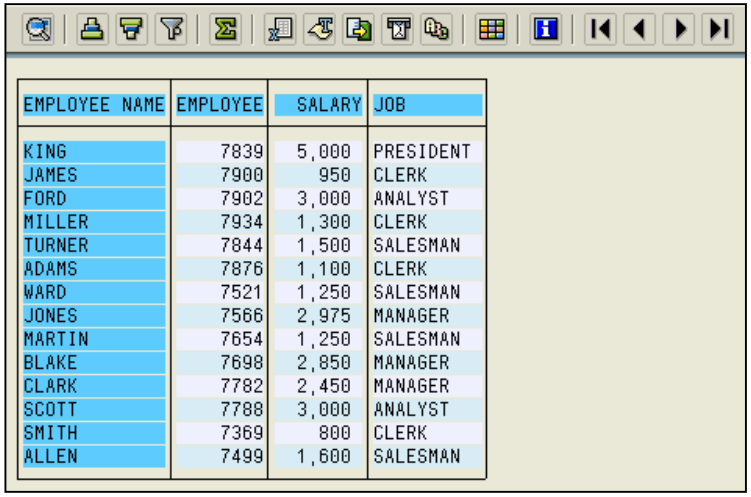
Simple Report

REUSE_ALV_GRID_DISPLAY: To display the results in grid rather than as a list.

- Parameters : same as reuse_alv_list_display

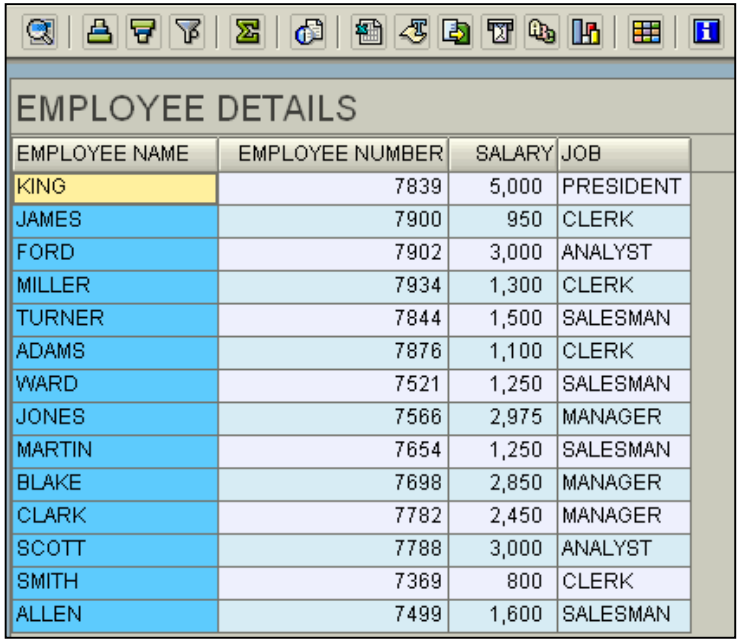
Display Output Internal Table

RESUSE_ALV_LIST_DISPLAY



EMPLOYEE NAME	EMPLOYEE	SALARY	JOB
KING	7839	5,000	PRESIDENT
JAMES	7900	950	CLERK
FORD	7902	3,000	ANALYST
MILLER	7934	1,300	CLERK
TURNER	7844	1,500	SALESMAN
ADAMS	7876	1,100	CLERK
WARD	7521	1,250	SALESMAN
JONES	7566	2,975	MANAGER
MARTIN	7654	1,250	SALESMAN
BLAKE	7698	2,850	MANAGER
CLARK	7782	2,450	MANAGER
SCOTT	7788	3,000	ANALYST
SMITH	7369	800	CLERK
ALLEN	7499	1,600	SALESMAN

REUSE_ALV_GRID_DISPLAY



EMPLOYEE DETAILS

EMPLOYEE NAME	EMPLOYEE NUMBER	SALARY	JOB
KING	7839	5,000	PRESIDENT
JAMES	7900	950	CLERK
FORD	7902	3,000	ANALYST
MILLER	7934	1,300	CLERK
TURNER	7844	1,500	SALESMAN
ADAMS	7876	1,100	CLERK
WARD	7521	1,250	SALESMAN
JONES	7566	2,975	MANAGER
MARTIN	7654	1,250	SALESMAN
BLAKE	7698	2,850	MANAGER
CLARK	7782	2,450	MANAGER
SCOTT	7788	3,000	ANALYST
SMITH	7369	800	CLERK
ALLEN	7499	1,600	SALESMAN

Demo

ALV List Display and ALV Grid Display

Field Catalog

A field catalog is prepared using the internal table (I_FIELDCAT) of type SLIS_T_FIELDCAT_ALV

Field catalog containing descriptions of the list output fields (usually a subset of the internal output table fields)

A field catalog is required for every ALV list output to add desired functionality (i.e. Key, Hotspot, Specific headings, Justify, Col. position etc) to certain fields of the output.

If not mentioned specifically, then the defaults are taken

No of ways to build Field Catalog

Preparing the data dictionary structure

Build field catalog using function module

- Reuse_alv_fieldcatlog_merge

Prepare the field Catalog manually

Demo

Create a Field Catalog Manually and use it in list display

OO ALV

The Control Framework is required for OO ALV as it provides global classes for various functionalities.

CL_GUI_ALV_GRID

- The wrapper class implemented to encapsulate ALV Grid functionality for list display.

Basic Components

While preparing a list to be displayed via an ALV grid control, we have some basic components to prepare. These are:

List data :

- Data in an internal table to be listed

Field Catalog :

- Define specifications on how the fields of our list will be displayed
- Has technical and additional information about display options for each column to be displayed.
- The internal table for the field catalog must be referenced to the dictionary type LVC_T_FCAT.

Container

- Storage area where the list will be displayed.
- It should be of type CL_GUI_CUSTOM_CONTAINER

Basic Components

Layout Structure :

- Fill a structure to specify general layout options for the grid
- To set
  - general display options
  - grid customizing
  - totals options
  - color adjustments etc...
- The layout structure must be of type LVC_S_LAYO.

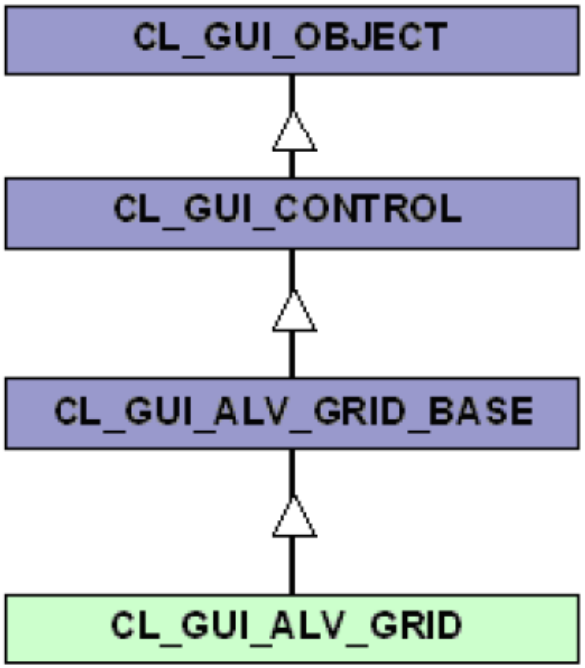
Basic Components

Event Handler

- To handle events triggered by the ALV Grid instance.
- Upon creating ALV Grid instance, register an instance of this event handler class to handle ALV Grid events
- Various Events are as follows
  - Print_Top_Of_Page:
    - Used for Headers. Handler is 'SET HANDLER'.
  - Print_End_Of_Page
    - Used for Footers. Handler is 'SET HANDLER'.
  - OnDropComplete
    - Event to change the state after a successful drag and drop operation.
  - OnDrag
    - To 'fetch' information from the drag source.
  - OnDrop
    - Used to use the dragged information in combination with drop source. Here, it should be checked whether the operation is successful

Grid Control – Inheritance Hierarchy

CL_GUI_ALV_GRID' class encapsulates communication with the instance on the presentation server, along with many other functions. For this reason, you should instantiate this class, not its super class.



Steps to work with OO ALV

Create an object of class CL_GUI_CUSTOM_CONTAINER.

Create an object of class CL_GUI_ALV_GRID.

Populate the internal table to display on the GRID.

Call the screen that contains the CUSTOM CONTAINER, in which the list has to be displayed.

Call the screen.

Call the method SET_TABLE_FOR_FIRST_DISPLAY of class CL_GUI_ALV_GRID and pass the required parameters

CL_GUI_ALV_GRID

‘CL_GUI_ALV_GRID’ class provides various methods and Events

Building Field Catalog

There are 3 methods for doing this:

- Automatic generation
- Semi-automatic generation
- Manual generation

Structure of Field Catalog (LVC_T_FCAT)

FIELDNAME	Assign a field name of your output table to a row of field catalog
REF_FIELD	Must specify this if field name in the output table is not identical to the field name of the field in Data Dictionary
REF_TABLE	Must fill this field only if the output table described by the current entry in the field catalog has a corresponding entry in the DDIC
COL_POS	Sequence of the fields
OUTPUTLEN	Desired width of the field in output
SCRTEXT_L/M/S	Field Labels

Building Field Catalog (Manually)

The work in this procedure is just filling the internal table for the field catalog. We have already seen the structure of a field catalog. To achieve filling the field catalog correctly, one must at least fill the above fields of the field catalog structure for each column of the list.

Output table fields *with* DDIC reference	Output table fields *without* DDIC reference	Explanation
FIELDNAME	FIELDNAME	Name of the field of the internal output table
REF_TABLE		Name of the DDIC reference structure
REF_FIELD		Name of the DDIC reference field (only needed if other than FIELDNAME)
	INTTYPE	ABAP data type of the field of the internal output table
	OUTPUTLEN	Column width
	COLTEXT	Column header
	SELTEXT	Column description in column selection for layout

Building Field Catalog (Manually)

```
DATA LS_FCAT TYPE LVC_S_FCAT .
LS_FCAT-FIELDNAME = 'CARRID‘.
LS_FCAT-INTTYPE = 'C'.
LS_FCAT-OUTPUTLEN = '3'.
LS_FCAT-COLTEXT = 'CARRIER ID'.
LS_FCAT-SELTEXT = 'CARRIER ID'.
APPEND LS_FCAT TO PT_FIELDCAT.
CLEAR LS_FCAT.
LS_FCAT-FIELDNAME = 'CONNID'.
LS_FCAT-REF_TABLE = 'SFLIGHT'.
LS_FCAT-REF_TABLE = 'CONNID'.
LS_FCAT-OUTPUTLEN = '3'.
LS_FCAT-COLTEXT = 'CONNECTION ID'.
LS_FCAT-SELTEXT = 'CONNECTION ID'.
APPEND LS_FCAT TO PT_FIELDCAT .
```

…. AND SO ON FOR ALL THE FIELDS TO BE DISPLAYED IN THE List

Demo

Create a Field Catalog Manually and use it in list display

Building Field Catalog (Semi-automatically)

Layout Adjustment

It comes now painting our ALV Grid in a general aspect. To define general appearance of our ALV Grid we fill a structure of type “LVC_S_LAYO”. This table contains fields and functionalities serviced by this adjustment. Some of the generally used options are as below:

ZEBRA	If this field is set, the list shows a striped pattern in the print preview and when it is printed (SPACE, 'X‘)
SMALLTITLE	If this field is set, the title size in the grid control is set to the font size of the column header. (SPACE, 'X‘)

```
FORM prepare_layout  CHANGING P_GS_LAYOUT TYPE lvc_s_layo.

P_GS_LAYOUT-zebra = 'X'.
P_GS_LAYOUT-grid_title = 'Flights'.
P_GS_LAYOUT-smalltitle = 'X'.

ENDFORM.                    " prepare_layout
```

ALV Display

Data transfer to the ALV control takes place during the call of method “SET_TABLE_FOR_FIRST_DISPLAY” of class “CL_GUI_ALV_GRID”. The method call must be programmed at the PBO event of the screen with the SAP Grid Control container.

Remember to use Pattern > ABAP Objects > Method of a Class

ALV Display

If the ALV_GRID is initial (First Call) the method "SET_TABLE_FOR_FIRST_DISPLAY" is called as described in the previous slide. Else on subsequent calls; "REFRESH_TABLE_DISPLAY" is called. Reason being; there is no need to instantiate the Custom Container, Grid every time in the PBO of the Screen.

The parameters of this method:

- IS_STABLE: If the row or column field of this structure is set, the position of the scroll bar for the rows or columns remains stable.
- I_SOFT_REFRESH: If set, any totals created, any sort order defined and any filter set for the data displayed remain unchanged when the grid control is refreshed.

```
    CALL METHOD GR_ALVGRID->REFRESH_TABLE_DISPLAY
*      EXPORTING
*     IS_STABLE      =
*     I_SOFT_REFRESH =
      EXCEPTIONS
        FINISHED       = 1
        OTHERS         = 2.
    IF SY-SUBRC <> 0.
      MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
                 WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
    ENDIF.
```

Setting Sort Condition

It is possible to set sort conditions for the table data. This is achieved by filling an internal table of structure “LVC_T_SORT” which consists of the sort criteria. To have an initial sorting, pass it to the parameter “IT_SORT” of the method “SET_TABLE_FOR_FIRST_DISPLAY”.

```
FORM PREPARE_SORT_TABLE  CHANGING PT_SORT TYPE LVC_T_SORT.

  DATA LS_SORT TYPE LVC_S_SORT.

  LS_SORT-SPOS = '1' .
  LS_SORT-FIELDNAME = 'CARRID'.
  LS_SORT-UP = 'X'. "A to Z
  LS_SORT-DOWN = SPACE.
  APPEND LS_SORT TO PT_SORT.

  LS_SORT-SPOS = '2'.
  LS_SORT-FIELDNAME = 'SEATSOCC'.
  LS_SORT-UP = SPACE.
  LS_SORT-DOWN = 'X'. "Z to A
  APPEND LS_SORT TO PT_SORT.

ENDFORM.                    " PREPARE_SORT_TABLE
```

Setting Filter Condition

The procedure is like the one in sorting. Here, the type of the table you must fill is “LVC_T_FILT”.

Filling this table is similar to filling a RANGES variable.

```
FORM PREPARE_FILTER_TABLE  CHANGING PT_FILT TYPE LVC_T_FILT.

  DATA LS_FILT TYPE LVC_S_FILT.

  LS_FILT-FIELDNAME = 'FLDATE'.
  LS_FILT-SIGN = 'E'.
  LS_FILT-OPTION = 'BT'.
  LS_FILT-LOW = '20030101'.
  LS_FILT-HIGH = '20031231'.
  APPEND LS_FILT TO PT_FILT.

ENDFORM.                    " PREPARE_FILTER_TABLE
```

Event Based Functionality

Additional Event based Functionalities that the ALV Grid can handle:-

- GENERAL SCHEME FOR THE EVENT HANDLER CLASS
- HOTSPOT CLICKING
- DOUBLE CLICKING
- PUSHBUTTONS ON THE LIST
- ADDING YOUR OWN FUNCTIONS
- OVERRIDING STANDARD FUNCTIONS
- MAKING ALV GRID EDITABLE
- CONTROLLING DATA CHANGES
- LINKING F1 HELP TO FIELDS
- LINKING F4 HELP TO FIELDS

Summary

In this lesson, you have learnt:

- Control Framework
- ALV Grid
- Non Event Based Functionality
- Event Based Functionality

Review Question

Question 1: _____ is used to find the length of the string.

Question 2: Condense and Concatenate command perform the same function.

- True/False