Ask a Question   Write a Blog Post   Login

Nagaraju Nadipudi

July 16, 2018   5 minute read

SAP List Viewer with Integrated Data Access (ALV with IDA)

Contents

Overview

ALV with IDA (SAP List Viewer with Integrated Data Access) helps tables that contain very large quantities of data to be displayed on the UI. The results of operations such as sorting, grouping, or filtering are also delivered with a very fast response time. It uses the in-memory database, such as SAP HANA, without having to switch to a new programming environment. There is no change in the User interface and standard functions (also ALV services). ALV functions are adjusted to the use of in-memory databases. The new general programming model (Coding Pushdown) is also optimally supported when using in-memory databases.

Examples

Example#1: Display Sales Order Details

Following example demonstrates how to display sales order with SALV IDA.

Output:

Check DB Capabilities
CHECK cl_salv_gui_table_ida=>db_capabilities( )->is_table_supported(iv_ddic_table_name = 'VBAK').

Create IDA
DATA(o_ida) = cl_salv_gui_table_ida=>create(iv_table_name = 'VBAK').

Set Maximum Rows Recommended

rows_recommended( ).

o_ida->set_maximum_number_of_rows(iv_number_of_rows = 2000).

ENDIF.

Display
o_ida->fullscreen( )->display( ).

Program

```abap
REPORT zr_hanaida_01.
" Check DB Capabilities
CHECK cl_salv_gui_table_ida=>db_capabilities( )->is_table_supported( iv_ddic_table_name = 'VBAK').

" Create ALV Grid
DATA(o_ida) = cl_salv_gui_table_ida=>create( iv_table_name = 'VBAK' ).

" Maximum Rows recommended
IF cl_salv_gui_table_ida=>db_capabilities( )->is_max_rows_recommended( ).
  o_ida->set_maximum_number_of_rows(  iv_number_of_rows = 2000 ).
ENDIF.

" Display
o_ida->fullscreen( )->display( ).
```

Output

Menu

Basic Example: Get list of Sales Order

Sales D	Created on	Time	Created	Valid	Valid	Doc. Date	Doc	TrG	SaTy	Ord	Guarant	Coll.	Dl	BB	Net val	Curr.	SOrg.	DChl	Dv	SG
1	02.06.2016	22:49:20	SFINUSER6			02.06.2016	C	0	OR						5.000,00	INR	BP01	01	01	
2	03.06.2016	15:48:18	SFINUSER6			03.06.2016	C	0	OR						2.000,00	INR	BP01	01	01	
3	11.04.2017	14:03:48	USER2			11.04.2017	C	0	OR						2.000,00	INR	BP01	01	01	
4	11.04.2017	14:05:30	USER2			11.04.2017	C	0	OR						2.000,00	INR	BP01	01	01	
5	11.04.2017	14:07:34	USER2			11.04.2017	C	0	OR						2.000,00	INR	BP01	01	01	
6	11.04.2017	19:57:49	USER2			11.04.2017	C	0	OR						500,00	INR	BP01	01	01	
7	11.04.2017	19:57:50	USER2			11.04.2017	C	0	OR						500,00	INR	BP01	01	01	
8	11.04.2017	19:57:48	USER2			11.04.2017	C	0	OR						0,00	INR	BP01	01	01	
9	11.04.2017	20:05:31	USER2			11.04.2017	C	0	OR						2.000,00	INR	BP01	01	01	
10	11.04.2017	20:07:20	USER2			11.04.2017	C	0	OR						2.000,00	INR	BP01	01	01	
11	11.04.2017	20:08:37	USER2			11.04.2017	C	0	OR						2.000,00	INR	BP01	01	01	
12	11.04.2017	20:12:01	USER2			11.04.2017	C	0	OR						2.000,00	INR	BP01	01	01	
13	11.04.2017	19:57:49	USER2			11.04.2017	C	0	OR						500,00	INR	BP01	01	01	
14	11.04.2017	20:14:43	USER2			11.04.2017	C	0	OR						2.000,00	INR	BP01	01	01	
15	11.04.2017	20:16:18	USER2			11.04.2017	C	0	OR						2.000,00	INR	BP01	01	01	

Example#2: Apply Filtering

Following example demonstrates how to set filter by using **PARAMETERS** and **SELECT-OPTIONS**

Step#1: Define Selection Screen

Step#2: Convert Parameters and Select-Options to Range table using CL_SALV_RANGE_TAB_COLLECTOR

Step#3: Apply filtering in ALV IDA

Selection Screen

```
DATA:

DATA:
 lr_vkorg TYPE RANGE OF vkorg.


PARAMETERS:
 p_vkorg TYPE vkorg.
 SELECT-OPTIONS:
   s_erdat FOR lv_erdat,
   s_kunnr FOR lv_kunnr.
```

Set Filter

```
" Set select Options
INSERT VALUE #( sign = 'I'
                option = 'EQ'
                low = p_vkorg
) INTO TABLE lr_vkorg.

" Range Table Collector
DATA(o_sel) = NEW cl_salv_range_tab_collector( ).
o_sel->add_ranges_for_name( iv_name = 'VKORG' it_ranges = lr_vkorg[] ).
o_sel->add_ranges_for_name( iv_name = 'ERDAT' it_ranges = s_erdat[] ).
o_sel->add_ranges_for_name( iv_name = 'KUNNR' it_ranges = s_kunnr[] ).

" Get name and ranges
o_sel->get_collected_ranges( IMPORTING et_named_ranges = DATA(lt_named_ranges) ).

" Set selected ranges to ALV
o_ida->set_select_options( it_ranges = lt_named_ranges ).
```

Output:

Selection Screen

Output

Example#3: ALV Display Setting

- List of fields to be displayed
- Set Sort Order
- Display Options
- Set Currency Reference Field

Restrict list of fields to be displayed
Step#1 Prepare internal table with list of fields in sorted order

Step#2: Pass the list of fields to SAL IDA

```
                    ( CONV string('KUNNR') )
                    ( CONV string('NETWR') )
                    ( CONV string('SPART') )
                    ( CONV string('VBELN') )
                    ( CONV string('VKORG') )
                    ( CONV string('VTWEG') )
                    ( CONV string('WAERK') ) ).

    o_ida->field_catalog( )->set_available_fields( EXPORTING its_field_names = lt_field_names ).
```

Output:

ALV IDA: Fields to be displayed

Sales D...	Created on	Net val...	Curr.	SOrg.	DChl	Dv	Sold-to...
1	02.06.2016	5.000,00	INR	BP01	01	01	1
2	03.06.2016	2.000,00	INR	BP01	01	01	1
3	11.04.2017	2.000,00	INR	BP01	01	01	1
4	11.04.2017	2.000,00	INR	BP01	01	01	1
5	11.04.2017	2.000,00	INR	BP01	01	01	1
6	11.04.2017	500,00	INR	BP01	01	01	1
7	11.04.2017	500,00	INR	BP01	01	01	1
8	11.04.2017	0,00	INR	BP01	01	01	1
9	11.04.2017	2.000,00	INR	BP01	01	01	1
10	11.04.2017	2.000,00	INR	BP01	01	01	1
11	11.04.2017	2.000,00	INR	BP01	01	01	1
12	11.04.2017	2.000,00	INR	BP01	01	01	1
13	11.04.2017	500,00	INR	BP01	01	01	1
14	11.04.2017	2.000,00	INR	BP01	01	01	1
15	11.04.2017	2.000,00	INR	BP01	01	01	1

Set Sort Order

Step#1: Prepare table for fields for which sorting is needed

Step#2: Pass the sorted table to SALV IDA

```
                                          da=>yt_sort_rule.
INSERT VALUE #( field_name = 'KUNNR'
                descending = abap_true
                is_grouped = abap_true
) INTO TABLE lt_sort_order.
INSERT VALUE #( field_name = 'WAERK'
                descending = abap_true
                is_grouped = abap_true
) INTO TABLE lt_sort_order.

" Set Sort Order
o_ida->default_layout( )->set_sort_order( EXPORTING it_sort_order = lt_sort_order ).
```
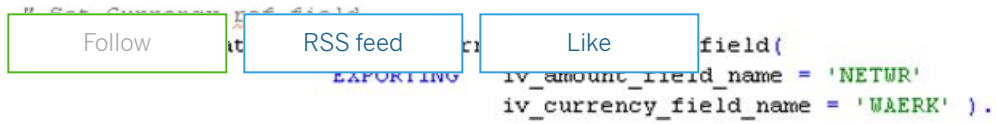
Output

Display Options
Set Currency Reference field
Set the reference fields for currency and quantity fields

```
" Set Currency ref field
         it         r   field(
         EXPORTING    iv_amount_field_name = 'NETWR'
                      iv_currency_field_name = 'WAERK' ).
```

Example#4: Event Handling

User interactions on the ALV can be handled by implementing the event handler methods.

- Double Click
- Toolbar Button

Double Click
Step#1: Define Event Handler Class/Method

Step#2: Implement Event Handler Class/Method

Step#3: Enable Double Click for SALV Grid

Step#4: Set Event Handler Method

Implement Event handle method

Implement Event handle method

```abap
PUBLIC SECTION.
    METHODS:
        constructor IMPORTING io_ida TYPE REF TO if_salv_gui_table_ida,
        handle_dbclick FOR EVENT  double_click OF if_salv_gui_table_display_opt.
    DATA:
        o_ida TYPE REF TO if_salv_gui_table_ida.
ENDCLASS.
CLASS lcl_handle_action IMPLEMENTATION.
    METHOD constructor.
        o_ida = io_ida.
    ENDMETHOD.
    METHOD handle_dbclick.
        DATA: ls_vbak         TYPE vbak,
              lr_vbeln        TYPE RANGE OF vbeln,
              lt_named_ranges TYPE if_salv_service_types=>yt_named_ranges.
        IF o_ida IS BOUND.
          IF o_ida->selection( )->is_row_selected( ).
            " Get selected row
            o_ida->selection( )->get_selected_row( IMPORTING es_row = ls_vbak ).
            DATA(o_ida_item) = cl_salv_gui_table_ida=>create( iv_table_name = 'VBAP' ).
            IF o_ida_item IS BOUND.
              INSERT VALUE #( name = 'VBELN'
                              sign = 'I'
                              option = 'EQ'
                              low = ls_vbak-vbeln ) INTO TABLE lt_named_ranges.
              o_ida_item->set_select_options( it_ranges = lt_named_ranges ).
              o_ida_item->fullscreen( )->display( ).
            ENDIF.
          ELSE.
            MESSAGE 'No row selected' TYPE 'I'.
          ENDIF.
        ENDIF.
    ENDMETHOD.
```

Enable Double Click and Set Selection Mode
Instantiate Event Handler Class and Set Handler
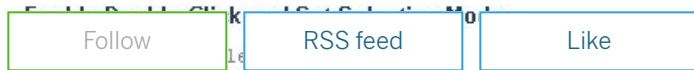
```
o_ida->display_options( )->enable_double_click( ).
" Set selection Mode
o_ida->selection( )->set_selection_mode( EXPORTING iv_mode = 'SINGLE' ).
```

Instantiate Event Handler Class and Set Handler

```
" Event Handler
DATA(o_handler) = NEW lcl_handle_action( io_ida = o_ida ).
SET HANDLER o_handler->handle_dbclick FOR o_ida->display_options( ).
```

Output
Toolbar Button
Create own buttons on toolbar
Step#1: Define Event Handler Class/Method

Step#2: Implement Event Handler Class/Method

Step#3: Add button to the Toolbar

Step#4: Enable Selection mode for SALV Grid

Step#4: Set Event Handler Method

1. Toolbar Event Handler Method Definition
2. Toolbar Event Handler Method Implementation

Toolbar Event Handler Method Definition

```
    F if_salv_gui_toolbar_ida,
```

Toolbar Event Handler Method Implementation

```
METHOD handle_action.
  DATA: ls_vbak        TYPE vbak,
        lr_vbeln       TYPE RANGE OF vbeln,
        lt_named_ranges TYPE if_salv_service_types=>yt_named_ranges.
  IF o_ida IS BOUND.
    IF o_ida->selection( )->is_row_selected( ).
      " Get selected row
      o_ida->selection( )->get_selected_row( IMPORTING es_row = ls_vbak ).
      DATA(o_ida_item) = cl_salv_gui_table_ida=>create( iv_table_name = 'VBAP' ).
      IF o_ida_item IS BOUND.
        INSERT VALUE #( name = 'VBELN'
                        sign = 'I'
                        option = 'EQ'
                        low = ls_vbak-vbeln ) INTO TABLE lt_named_ranges.
        o_ida_item->set_select_options( it_ranges = lt_named_ranges ).
        o_ida_item->fullscreen( )->display( ).
      ENDIF.
    ELSE.
      MESSAGE 'No row selected' TYPE 'I'.
    ENDIF.
  ENDIF.
```

3.Add Button on to the Toolbar
4.Set Event Handler Method

Add Button on to the Toolbar

```
" Toolbar
o_ida->toolbar( )->add_button( EXPORTING iv_fcode = 'DISP' iv_text = 'Display' ).
```

Set Event Handler Method

```
SET HANDLER o_handler->handle_action FOR o_ida->toolbar( ).
```

Output
Click on Display Button

			Sold-to par...

▽1400000602 (1) [Sold-to party]
  ▽EUR (1) [Document Currency]

2840	10.03.2018	100,00 EUR	ZIT1	10	01	1400000602

▷ 1400000597 (2) [Sold-to party]
▷ 1400000595 (4) [Sold-to party]
▷ 1400000594 (1) [Sold-to party]
▷ 1400000593 (1) [Sold-to party]
▷ 1400000567 (5) [Sold-to party]
▷ 1400000566 (1) [Sold-to party]
▷ 1400000563 (19) [Sold-to party]
▽1400000562 (22) [Sold-to party]
  ▽EUR (22) [Document Currency]

2464	09.02.2018	4.905,00 EUR	ZIT1	10	01	1400000562
2467	09.02.2018	1.820,00	ZIT1	10	01	
2484	09.02.2018	4.100,00	ZIT1	10	01	
2497	09.02.2018	8.370,00	ZIT1	10	01	

Click on Display Button

Sales Orders

Sales D	Ite	Mater	MatEnter	Pr.Ref	Bat	Matl Gro	Description	ItCa	IT	D	Bil	HgL	AltI	Rj	Prod.hi	Targ. V	Target q	UoM	ConvFac	ConvFac
2464	10	3975	3975			05	Cozze	TAN		A	0	0				0,00	0	PC	1	
2464	20	3811	3811			05	Borsa nera	TAN		A	0	0				0,00	0	PC	1	
2464	30	3789	3789			05	Borsa Bianca	TAN		A	0	0				0,00	0	PC	1	

Example#5: Calculation Fields

We can also display calculated fields apart from the Standard database table fields using calculation fields

Implement Calculation field Handler
Interface: IF_SALV_IDA_CALC_FIELD_HANDLER

Step#1: Define Structure with custom fields.

In our case we have two calculation fields namely TAXAMOUNT and NETAMOUNT

Step#2: Implement method to get calculation field structure

Register Structure type to the SALV IDA Grid

What are the database fields needed to calculate the custom calculation fields

Step#4: Implement method to calculate Line

Perform the Calculation

```
CLASS lcl_cal_fields DEFINITION.
  PUBLIC SECTION.
    INTERFACES if_salv_ida_calc_field_handler.
    TYPES:
      BEGIN OF ty_custom_fields,
        taxamount TYPE netwr,
        netamount TYPE netwr,
      END OF ty_custom_fields.
  PROTECTED SECTION.
  PRIVATE SECTION.
ENDCLASS.
CLASS lcl_cal_fields IMPLEMENTATION.
  METHOD if_salv_ida_calc_field_handler~get_calc_field_structure.
    ro_calc_field_structure ?= cl_abap_typedescr=>describe_by_name( 'TY_CUSTOM_FIELDS' ).
  ENDMETHOD.
  METHOD if_salv_ida_calc_field_handler~get_requested_fields.
    " Specify Source field for Calculating the Custom Field
    rts_db_field_name = VALUE #( ( CONV string('NETWR') ) ). " Tax Amount
    rts_db_field_name = VALUE #( ( CONV string('NETWR') ) ). " Net Amount
  ENDMETHOD.
  METHOD if_salv_ida_calc_field_handler~calculate_line.
    " Calculate Values
    DATA:
      ls_vbak       TYPE vbak,
      ls_cal_fields TYPE ty_custom_fields.
    ls_vbak = is_data_base_line.
    ls_cal_fields-taxamount = ls_vbak-netwr * '0.10'.
    ls_cal_fields-netamount = ls_vbak-netwr -  ls_cal_fields-taxamount.
    es_calculated_fields = ls_cal_fields.
  ENDMETHOD.
  METHOD if_salv_ida_calc_field_handler~end_page.
  ENDMETHOD.
  METHOD if_salv_ida_calc_field_handler~start_page.
  ENDMETHOD.
ENDCLASS
```

Step#5: Pass the calculation fields handler object to SALV IDA Create method

Step #5: Pass the calculation field handler object to ALV method

```
" Calculation Fields
DATA(o_ida) = cl_salv_gui_table_ida=>create(
                      iv_table_name = 'VBAK'
                      io_calc_field_handler = NEW lcl_cal_fields( ) ).
```

Output:

Example#6: Working with CDS Views
We can use CDS Views as a data source to display data using SALV IDA

CDS Views

```
@EndUserText.label: 'Customers'
define view ZCDS_CUSTOMER as select from kna1
{
    key kunnr,
    name1,
    name2,
    ort01,
    land1
}
```

ALV Program

```
START-OF-SELECTION.
    DATA(o_ida) =  cl_salv_gui_table_ida=>create_for_cds_view(
    iv_cds_view_name = 'ZCDS_CUSTOMER'  ).
    " Display
    o_ida->fullscreen( )->display( ).
```

Output

ALV IDA: CDS Views

Customer	Name 1	Name 2	City	Cty
1	SFIN CUSTOMERS			IN
6	Mario Bianchi			IT
7	carlotta test			IT
1400000000	carlotta test2			IT
1400000001	gio			IT
1400000002	test2			IT
1400000003	acc1			IT
ADC	gio		81030	IT
CUSTOMER	gio		81030	IT
GIOVANNI	giovanni cristiano		81030	IT
GIOVANNI2	giovanni cristiano		81030	IT
1400000004	CLIENTE NAZIONALE		roma	IT
1500000001	Gianluca		Giugliano in Campania	IT
1500000006	DEST 1		Battipaglia	IT
1400000005	Casu Matteo		Tortoli	IT
203	Astolfo	Gilberto	Salerno	IT
1400000306	divano srl		ariano	IT
1400000008	Simone Uiip Spa		Marzano di Nola	IT

Example#7: Authority Check

Authority check can be applied when working with SALV IDA. We have to use the method **ADD_AUTHORIZATION_FOR_OBJECT** of the interface

B

```
" Authorization Check
o_ida->add_authorization_for_object( EXPORTING
iv_authorization_object = 'ZVKORG'
it_activities =
    VALUE if_salv_gui_types_ida=>yt_activities( ( auth_field = 'ACTVT' value = '03' ) )
it_field_mapping =
    VALUE if_salv_gui_types_ida=>yt_field_mapping( ( auth_field  = 'ZVKORG' view_field = 'VKORG' ) )
).
```

Classes/Interfaces:

Class/Interface	Purpose
IF_SALV_GUI_TABLE_IDA	ALV with integrated data access (ALV with IDA)
CL_SALV_GUI_TABLE_IDA	ALV with integrated data access (ALV with IDA)
CL_SALV_RANGE_TAB_COLLECTOR	Class for collecting range tables for multiple fields
IF_SALV_GUI_TYPES_IDA	ALV IDA API : Public data types
IF_SALV_IDA_CALC_FIELD_HANDLER	IDA API: Handler for Calculated Fields
IF_SALV_GUI_TABLE_DISPLAY_OPT	IDA ALV: Table Display Options API
IF_SALV_GUI_TOOLBAR_IDA	IDA API Toolbar Manipulation Object

Reference

https://help.sap.com/doc/saphelp_scm700_ehp03/7.0.3/en-US/f0/1df4a12b0c41c59fc72d007915ea43/frameset.htm

Alert Moderator

Assigned tags

ABAP Development | SAP SALV IDA |

Related Blog Posts

ALV IDA and HANA as a secondary database
By Wouter Lemaire , Dec 12, 2014

ALV IDA On External Dictionary View
By Amol Samte , Jul 02, 2015

Big Data on SAP UIs – Advances in ABAP List Viewer ALV on SAP HANA
By Former Member , Nov 25, 2014

Related Questions

Refresh ALV IDA with CDS
By Frank Gutknecht , May 09, 2016

ALV with ida access

By **Aditya Sharma** , Nov 26, 2019

By **Saurabh Gupta** , Sep 11, 2017

1 Comment

You must be **Logged on** to comment or reply to a post.

Inderpreet Singh

March 21, 2019 at 7:02 am

Hi,

We are using the ALV IDA in our project and we noticed that the layout button of the ALV list is placed on the extreme right as shown below:

and other icons are on the left side.
Our client wants to display all the buttons together on the left side.

Is there any way to align the layout button to left.

Thanks in advance!!!

Like (0)

Find us on

Privacy	Terms of Use
Legal Disclosure	Copyright
Trademark	Cookie Preferences
Newsletter	Support