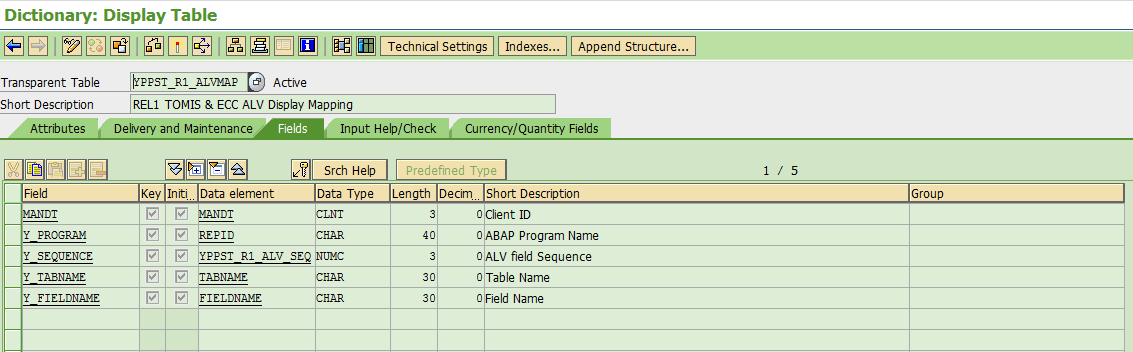
Dynamic ALV Report

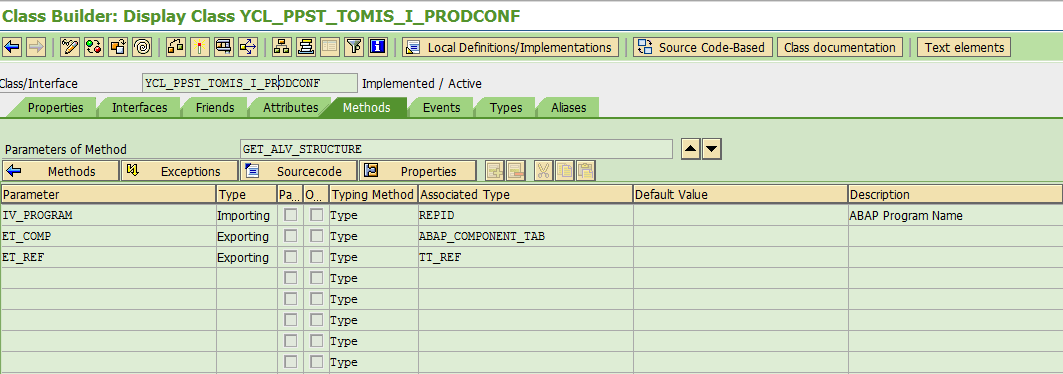
Table for dynamic alv



Class

Parameter

  types:  
    tt\_ref TYPE STANDARD TABLE OF salv\_s\_ddic\_reference WITH DEFAULT KEY .



ALV structure call

Data:       gt\_comp         TYPE cl\_abap\_structdescr=>component\_table,  
          gt\_ref          TYPE STANDARD TABLE OF salv\_s\_ddic\_reference,

    DATA: lo\_new\_type TYPE REF TO cl\_abap\_structdescr,  
          lo\_new\_tab  TYPE REF TO cl\_abap\_tabledescr,  
          lo\_data     TYPE REF TO data,  
          lo\_line     TYPE REF TO data.

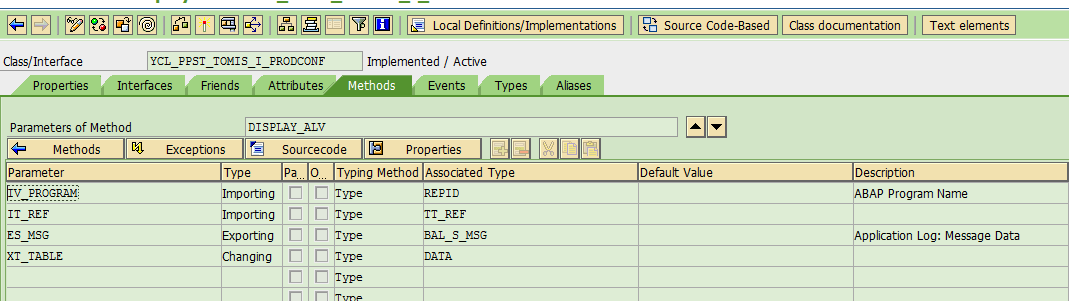
\* ALV Structure  
    ycl\_ppst\_tomis\_i\_prodconf=>get\_alv\_structure( EXPORTING iv\_program = sy-repid  
                  IMPORTING et\_comp   = gt\_comp  
                            et\_ref    = gt\_ref ).  
    IF  gt\_comp[] IS NOT INITIAL.  
      lo\_new\_type = cl\_abap\_structdescr=>create( gt\_comp ).  
  
      lo\_new\_tab = cl\_abap\_tabledescr=>create(  
                   p\_line\_type  = lo\_new\_type  
                   p\_table\_kind = cl\_abap\_tabledescr=>tablekind\_std  
                   p\_unique     = abap\_false ).  
  
      CREATE DATA lo\_data TYPE HANDLE lo\_new\_tab.  
\*  
\* 6. New internal table in the fieldsymbols  
      ASSIGN lo\_data->\* TO <gt\_alv>.  
      CREATE DATA lo\_line LIKE LINE OF <gt\_alv>.  
      ASSIGN lo\_line->\* TO <gs\_alv>.  
    ENDIF.

To Fill AVL

  METHOD fill\_alv .  
\* Description    : To fill the alv structure  
    FIELD-SYMBOLS <lv\_field> TYPE any.  
    IF is\_tomis IS NOT INITIAL.  
      MOVE-CORRESPONDING is\_tomis TO <gs\_alv>.  
      ASSIGN COMPONENT ycl\_ppst\_tomis\_i\_prodconf=>c\_process\_flag OF STRUCTURE <gs\_alv> TO <lv\_field>.  
      IF sy-subrc EQ 0.  
        <lv\_field> = ycl\_ppst\_tomis\_i\_prodconf=>c\_red.  
      ENDIF.  
      APPEND <gs\_alv> TO <gt\_alv>.  
    ENDIF.  
    IF is\_conf IS NOT INITIAL.  
      READ TABLE <gt\_alv> ASSIGNING <gs\_alv> WITH KEY (ycl\_ppst\_tomis\_i\_prodconf=>c\_meldung) = is\_conf-y\_id\_meldung  
                                                  (ycl\_ppst\_tomis\_i\_prodconf=>c\_vorgang) = is\_conf-y\_vorgang.  
      IF sy-subrc EQ 0.  
        MOVE-CORRESPONDING is\_conf TO <gs\_alv>.  
        ASSIGN COMPONENT ycl\_ppst\_tomis\_i\_prodconf=>c\_process\_flag OF STRUCTURE <gs\_alv> TO <lv\_field>.  
        IF sy-subrc EQ 0.  
          IF is\_conf-y\_conf\_flag EQ ycl\_ppst\_tomis\_i\_prodconf=>c\_process\_y.  
            <lv\_field> = ycl\_ppst\_tomis\_i\_prodconf=>c\_green.  
          ELSE.  
            <lv\_field> = ycl\_ppst\_tomis\_i\_prodconf=>c\_red.  
          ENDIF.  
        ENDIF.  
      ELSE.  
        READ TABLE me->gt\_r1\_tomis ASSIGNING FIELD-SYMBOL(<ls\_tomis>) WITH KEY y\_id\_meldung = is\_conf-y\_id\_meldung  
                                                                        y\_vorgang   =  is\_conf-y\_vorgang.  
        IF sy-subrc EQ 0.  
          MOVE-CORRESPONDING <ls\_tomis> TO <gs\_alv>.  
        ENDIF.  
        MOVE-CORRESPONDING is\_conf TO <gs\_alv>.  
        ASSIGN COMPONENT ycl\_ppst\_tomis\_i\_prodconf=>c\_process\_flag OF STRUCTURE <gs\_alv> TO <lv\_field>.  
        IF sy-subrc EQ 0.  
          IF is\_conf-y\_conf\_flag EQ ycl\_ppst\_tomis\_i\_prodconf=>c\_process\_y.  
            <lv\_field> = ycl\_ppst\_tomis\_i\_prodconf=>c\_green.  
          ELSE.  
            <lv\_field> = ycl\_ppst\_tomis\_i\_prodconf=>c\_red.  
          ENDIF.  
        ENDIF.  
        APPEND <gs\_alv> TO <gt\_alv>.  
      ENDIF.  
    ENDIF.  
  
  ENDMETHOD.

To Display ALV

  types:  
    tt\_ref TYPE STANDARD TABLE OF salv\_s\_ddic\_reference WITH DEFAULT KEY .



  METHOD display\_alv.  
    DATA: lo\_sort TYPE REF TO cl\_salv\_sorts.  
\*  Display ALV  
    TRY.  
\* Create object for ALV  
        CALL METHOD cl\_salv\_table=>factory(  
          IMPORTING  
            r\_salv\_table = DATA(lo\_alv)  
          CHANGING  
            t\_table      = xt\_table ).  
  
\*\*\*\*\*\*For Sorting the internal Table on 'Y\_ID\_MELDUNG'  
        lo\_sort = lo\_alv->get\_sorts( ).  
        TRY.  
            CALL METHOD lo\_sort->add\_sort  
              EXPORTING  
                columnname = c\_meldung  
                position   = 1.  
  
          CATCH cx\_salv\_not\_found ##no\_handler.  
          CATCH cx\_salv\_existing ##no\_handler.  
          CATCH cx\_salv\_data\_error ##no\_handler.  
        ENDTRY.  
\* Optimize columns  
        lo\_alv->get\_columns( )->set\_optimize( if\_salv\_c\_bool\_sap=>true ).  
        DATA(lo\_columns) = lo\_alv->get\_columns( ).  
        lo\_columns->set\_key\_fixation( value = abap\_true ).  
        lo\_alv->set\_screen\_status(  
              pfstatus      =  ycl\_ppst\_tomis\_i\_prodconf=>c\_yppst\_standard  
              report        =  iv\_program  
              set\_functions =  lo\_alv->c\_functions\_all ).  
  
        IF it\_ref  IS NOT INITIAL.  
          LOOP AT it\_ref ASSIGNING FIELD-SYMBOL(<ls\_ref>).  
            DATA(lo\_column) = lo\_columns->get\_column( <ls\_ref>-field ).  
            lo\_column->set\_ddic\_reference( <ls\_ref> ).  
          ENDLOOP.  
        ENDIF.  
        lo\_columns->set\_exception\_column( value = c\_process\_flag ).  
        lo\_alv->display( ).  
      CATCH cx\_salv\_msg INTO DATA(lo\_msg).  
  
        DATA(ls\_msg) = lo\_msg->get\_message( ).  
  
        IF ls\_msg IS NOT INITIAL .  
          es\_msg = ls\_msg.  
        ENDIF.  
    ENDTRY.  
  ENDMETHOD.