class ZCL\_SWF\_NOTI\_UTIL definition  
  public  
  final  
  create public .  
  
public section.  
  
  interfaces BI\_OBJECT .  
  interfaces BI\_PERSISTENT .  
  interfaces IF\_WORKFLOW .  
  
  class-methods READ\_LF\_APPROVER  
    importing  
      value(I\_AGENTS) type ZTT\_AGENTS optional  
      value(I\_SWRD) type ZST\_SWRD optional  
      value(I\_SEQ) type ZSEQ optional  
      value(I\_REQID) type TIM\_REQ\_ID optional  
      value(I\_REQ\_VER) type TIM\_REQ\_VERSION optional  
      value(I\_INIT) type SWP\_INITIA optional  
    exporting  
      value(E\_AGENT) type SWP\_AGENT  
      value(E\_FINALAPP) type ZSEQ  
      value(E\_PROCESS) type ZPROCESS .  
  class-methods NOTIFY  
    importing  
      value(I\_SWND) type ZST\_SWND optional  
      value(I\_AGENTS) type ZTT\_AGENTS optional  
    exporting  
      value(E\_RETURN) type BAPIRET2 .  
  class-methods GET\_STATUS  
    importing  
      value(I\_REQSTATUS) type TIM\_REQ\_STATUS  
    exporting  
      value(E\_REQSTATUS) type TIM\_REQ\_STATUS .  
protected section.  
private section.  
  
  types:  
    BEGIN OF ts\_usrd,  
      objid       TYPE hrobjid,  
      otype       TYPE otype,  
      user        TYPE sysid,  
      pernr       TYPE pernr\_d,  
      first\_name  TYPE vorna,  
      second\_name TYPE nach2,  
      last\_name   TYPE nachn,  
      full\_name   TYPE pad\_cname,  
      plans       TYPE plans,  
      plstx       TYPE string,  
      email       TYPE comm\_id\_long,  
      wf\_aty      TYPE char1,  
    END OF ts\_usrd .  
  types:  
    BEGIN OF ts\_notify\_d,  
      email\_addr   TYPE ad\_smtpadr,  
      subject      TYPE so\_obj\_des,  
      title\_attach TYPE so\_obj\_des,  
      main\_text    TYPE bcsy\_text,  
      addr\_name    TYPE ad\_smtpadr,  
      addr\_string  TYPE ad\_smtpadr,  
      pdf          TYPE fpformoutput,  
      attach\_text  TYPE ehfndt\_solisti1,  
      att          TYPE solix\_tab,  
    END OF   ts\_notify\_d .  
  types:  
    BEGIN OF ts\_objid,  
      objid TYPE hrobjid,  
      otype TYPE otype,  
      sclas TYPE sclas,  
      sobid TYPE sobid,  
    END OF  ts\_objid .  
  types:  
    tt\_usrd TYPE STANDARD TABLE OF ts\_usrd WITH DEFAULT KEY .  
  types:  
    tt\_objid TYPE HASHED TABLE OF ts\_objid WITH UNIQUE KEY objid otype .  
  
  class-data INITIATOR\_DET type TS\_USRD .  
  class-data APPROVER\_DET type TS\_USRD .  
  class-data NOTIFY\_D type TS\_NOTIFY\_D .  
  class-data DECISION\_NOTE type SO\_TEXT255 .  
  constants C\_CATID\_BO type SIBFCATID value 'BO' ##NO\_TEXT.  
  constants C\_CATID\_CL type SIBFCATID value 'CL' ##NO\_TEXT.  
  constants C\_ENDDA type ENDDA value '99991231' ##NO\_TEXT.  
  constants C\_OTYPE\_US type OTYPE value 'US' ##NO\_TEXT.  
  constants C\_OTYPE\_S type OTYPE value 'S' ##NO\_TEXT.  
  constants C\_OTYPE\_P type OTYPE value 'P' ##NO\_TEXT.  
  constants C\_OTYPE\_E type OTYPE value 'E' ##NO\_TEXT.  
  constants C\_SUBTY\_USER type SUBTY value '0001' ##NO\_TEXT.  
  constants C\_SUBTY\_EMAIL type SUBTY value '0010' ##NO\_TEXT.  
  constants C\_MODULE\_FI type ZMODULE value 'FI' ##NO\_TEXT.  
  constants C\_PROCESS\_PP type ZPROCESS value 'PP' ##NO\_TEXT.  
  constants C\_MODULE\_MM type ZMODULE value 'MM' ##NO\_TEXT.  
  constants C\_PROCESS\_PO type ZPROCESS value 'PO' ##NO\_TEXT.  
  constants C\_PROCESS\_PR type ZPROCESS value 'PR' ##NO\_TEXT.  
  constants C\_PROCESS\_MIR type ZPROCESS value 'MIR' ##NO\_TEXT.  
  constants C\_MODULE\_HCM type ZMODULE value 'HCM' ##NO\_TEXT.  
  constants C\_PROCESS\_TR type ZPROCESS value 'TR' ##NO\_TEXT.  
  constants C\_PROCESS\_TE type ZPROCESS value 'TE' ##NO\_TEXT.  
  constants C\_PROCESS\_LR type ZPROCESS value 'LR' ##NO\_TEXT.  
  constants C\_DECISION\_I type ZDECISION value 'I' ##NO\_TEXT.  
  constants C\_DECISION\_R type ZDECISION value 'R' ##NO\_TEXT.  
  constants C\_DECISION\_A type ZDECISION value 'A' ##NO\_TEXT.  
  constants C\_DECISION\_ALM type ZDECISION value 'ALM' ##NO\_TEXT.  
  constants C\_DECISION\_RLM type ZDECISION value 'RLM' ##NO\_TEXT.  
  constants C\_DECISION\_AR type ZDECISION value 'AR' ##NO\_TEXT.  
  constants C\_DECISION\_FI type ZDECISION value 'FI' ##NO\_TEXT.  
  constants C\_DECISION\_F type ZDECISION value 'F' ##NO\_TEXT.  
  constants C\_DECISION\_SBC type ZDECISION value 'SBC' ##NO\_TEXT.  
  constants C\_SENDER type AD\_SMTPADR value 'NDRMF SAP' ##NO\_TEXT.  
  constants C\_SENDER\_EMAIL type AD\_SMTPADR value 'sap.notifications@ndrmf.pk' ##NO\_TEXT.  
  constants C\_WI\_TYPE\_F type SWW\_WITYPE value 'F' ##NO\_TEXT.  
  constants C\_WI\_TYPE\_W type SWW\_WITYPE value 'W' ##NO\_TEXT.  
  constants C\_WF\_ATY\_I type CHAR1 value 'I' ##NO\_TEXT.  
  constants C\_WF\_ATY\_A type CHAR1 value 'A' ##NO\_TEXT.  
  constants C\_DECI\_NR type ZDNOTE value '01' ##NO\_TEXT.  
  constants C\_WI\_STAT\_READY type SWW\_WISTAT value 'READY' ##NO\_TEXT.  
  constants C\_WI\_STAT\_COMPLETED type SWW\_WISTAT value 'COMPLETED' ##NO\_TEXT.  
  constants C\_PROCESS\_LR2 type ZPROCESS value 'LR2' ##NO\_TEXT.  
  
  class-methods SEND\_MAIL  
    importing  
      value(I\_NOTIFY\_D) type TS\_NOTIFY\_D optional  
    exporting  
      value(ES\_RETURN) type BAPIRET2 .  
  class-methods HOLD\_AGENTS  
    importing  
      value(I\_SWND) type ZST\_SWND optional  
      value(I\_AGENTS) type ZTT\_AGENTS optional .  
  class-methods GET\_NOTIF\_DET  
    importing  
      value(I\_SWND) type ZST\_SWND optional  
    exporting  
      value(E\_USRD) type TT\_USRD .  
  class-methods HOLD\_DEC\_NOTE  
    importing  
      value(I\_SWND) type ZST\_SWND optional .  
  class-methods GET\_EMAIL\_CONTENT  
    importing  
      value(I\_SWND) type ZST\_SWND optional  
      value(I\_USRD) type TS\_USRD optional  
    exporting  
      value(E\_SUBJECT) type SO\_OBJ\_DES  
      value(E\_MAIN\_TEXT) type BCSY\_TEXT .  
  class-methods GET\_UMD  
    changing  
      value(C\_USRD) type TT\_USRD optional .  
ENDCLASS.  
  
  
  
CLASS ZCL\_SWF\_NOTI\_UTIL IMPLEMENTATION.  
  
  
\* <SIGNATURE>---------------------------------------------------------------------------------------+  
\* | Static Private Method ZCL\_SWF\_NOTI\_UTIL=>GET\_EMAIL\_CONTENT  
\* +-------------------------------------------------------------------------------------------------+  
\* | [--->] I\_SWND                         TYPE        ZST\_SWND(optional)  
\* | [--->] I\_USRD                         TYPE        TS\_USRD(optional)  
\* | [<---] E\_SUBJECT                      TYPE        SO\_OBJ\_DES  
\* | [<---] E\_MAIN\_TEXT                    TYPE        BCSY\_TEXT  
\* +--------------------------------------------------------------------------------------</SIGNATURE>  
    METHOD get\_email\_content.  
      DATA : startdate(10) TYPE c,  
             enddate(10)   TYPE c.  
\*\*&---------------------------------------------------------------------\*  
\*\*      Get instance number of workitem  
\*\*----------------------------------------------------------------------\*  
\*   Get top workflow TO id  
      SELECT SINGLE top\_wi\_id  
        FROM swwwihead  
        INTO @DATA(top\_wi\_id)  
          WHERE wi\_id EQ @i\_swnd-workitem.  
\*   Get instid  
      SELECT SINGLE instid  
      FROM sww\_wi2obj  
      INTO @DATA(instid)  
            WHERE wi\_id EQ @top\_wi\_id  
              AND catid EQ @c\_catid\_bo.  
  
\*      \*          Get id of request item list  
\*\*&---------------------------------------------------------------------\*  
\*\*   BOC  GETTING LEAVE REQUEST'S DATA  BY UMER EHSAN     5/29/2021  
\*\*----------------------------------------------------------------------\*  
      SELECT SINGLE \*  
        FROM ptreq\_header  
        INTO @DATA(ptreq\_header)  
         WHERE workitem\_id EQ @i\_swnd-workitem  
           AND  version\_no EQ '1'.  
\*           Get item\_id  
      SELECT SINGLE item\_ins  
        FROM ptreq\_items  
        INTO @DATA(item\_ins)  
        WHERE item\_list\_id = @ptreq\_header-item\_list\_id.  
      DATA(item\_id) = CONV tim\_req\_item\_id( item\_ins ).  
\*           Get request data  
      SELECT SINGLE \*  
        FROM ptreq\_attabsdata  
        INTO @DATA(ptreq\_attabsdata)  
        WHERE item\_id EQ @item\_id.  
  
\*      \*           Get note text  
      SELECT SINGLE notice\_text  
        FROM ptreq\_notice  
        INTO @DATA(notice\_text)  
        WHERE request\_id EQ @ptreq\_header-request\_id.  
\*           Get leave type  
      SELECT SINGLE atext  
        FROM t554t  
        INTO @DATA(leave\_text)  
        WHERE sprsl EQ @sy-langu  
          AND moabw EQ '98'         ##NO\_TEXT  
          AND awart EQ @ptreq\_attabsdata-subty.  
\*          Get quota type  
      SELECT SINGLE \*  
        FROM zlqty  
        INTO @DATA(ls\_zlqty)  
        WHERE subty EQ @ptreq\_attabsdata-subty.  
\*          Get quota detail  
      SELECT SINGLE \*  
        FROM pa2006  
        INTO @DATA(ls\_pa2006)  
        WHERE pernr EQ @ptreq\_attabsdata-pernr  
          AND subty EQ @ls\_zlqty-quotaty  
          AND ( begda LE @ptreq\_attabsdata-begda AND endda GE @ptreq\_attabsdata-begda ).  
      DATA(entitled) = ls\_pa2006-anzhl.  
      DATA(availed)  = ls\_pa2006-kverb.  
      DATA(balance)  = entitled - availed.  
      DATA(total\_days) = ptreq\_attabsdata-kaltg.  
\*          Call function to convert start date  
      CALL FUNCTION 'CONVERT\_DATE\_TO\_EXTERNAL'  
        EXPORTING  
          date\_internal            = ptreq\_attabsdata-begda  
        IMPORTING  
          date\_external            = startdate  
        EXCEPTIONS  
          date\_internal\_is\_invalid = 1  
          OTHERS                   = 2.  
\*          Call function to convert end date  
      CALL FUNCTION 'CONVERT\_DATE\_TO\_EXTERNAL'  
        EXPORTING  
          date\_internal            = ptreq\_attabsdata-endda  
        IMPORTING  
          date\_external            = enddate  
        EXCEPTIONS  
          date\_internal\_is\_invalid = 1  
          OTHERS                   = 2.  
  
  
"Commented by M Hamza Zahid 01.04.2022  
  
\*      IF ( ptreq\_attabsdata-subty EQ '0100' )  
\*      OR ( ptreq\_attabsdata-subty EQ '0200' )  
\*      OR ( ptreq\_attabsdata-subty EQ '0300' )  
\*      OR ( ptreq\_attabsdata-subty EQ '0600' ).  
\*          i\_swnd-process = 'LR'.  
  
"Commented by M Hamza Zahid 01.04.2022  
  
"Added by M Hamza Zahid 01.04.2022  
      IF ( ptreq\_attabsdata-subty EQ '0200' )  
      OR ( ptreq\_attabsdata-subty EQ '0300' )  
      OR ( ptreq\_attabsdata-subty EQ '0600' ).  
  
        i\_swnd-process = 'LR'.  
  
"Added by M Hamza Zahid 01.04.2022  
  
"Commented by M Hamza Zahid 01.04.2022  
  
\*      ELSEIF ( ptreq\_attabsdata-subty EQ '0400' )  
\*      OR ( ptreq\_attabsdata-subty EQ '0401' )  
\*      OR ( ptreq\_attabsdata-subty EQ '0700' )  
\*      OR ( ptreq\_attabsdata-subty EQ '0710' )  
\*      OR ( ptreq\_attabsdata-subty EQ '0720' )  
\*      OR ( ptreq\_attabsdata-subty EQ '0500' )  
\*      OR ( ptreq\_attabsdata-subty EQ '0550' )  
\*      OR ( ptreq\_attabsdata-subty EQ '0800' ).  
\*  
\*        i\_swnd-process = 'LR2'.  
  
"Commented by M Hamza Zahid 01.04.2022  
  
  
"Added by M Hamza Zahid 01.04.2022  
  
  
  
      ELSEIF ( ptreq\_attabsdata-subty EQ '0100' )  
      OR ( ptreq\_attabsdata-subty EQ '0400' )  
      OR ( ptreq\_attabsdata-subty EQ '0401' )  
      OR ( ptreq\_attabsdata-subty EQ '0700' )  
      OR ( ptreq\_attabsdata-subty EQ '0710' )  
      OR ( ptreq\_attabsdata-subty EQ '0720' )  
      OR ( ptreq\_attabsdata-subty EQ '0500' )  
      OR ( ptreq\_attabsdata-subty EQ '0550' )  
      OR ( ptreq\_attabsdata-subty EQ '0800' ).  
  
        i\_swnd-process = 'LR2'.  
  
  
"Added by M Hamza Zahid 01.04.2022  
  
  
      ENDIF.  
\*\*&---------------------------------------------------------------------\*  
\*\*   EOC  GETTING LEAVE REQUEST'S DATA BY UMER EHSAN 5/29/2021  
\*\*----------------------------------------------------------------------\*  
  
  
\*\*&---------------------------------------------------------------------\*  
\*\*      Email content of all modules  
\*\*----------------------------------------------------------------------\*  
      CASE i\_swnd-module.  
\*\*&---------------------------------------------------------------------\*  
\*\*       Module HCM  
\*\*----------------------------------------------------------------------\*  
        WHEN c\_module\_hcm.                     "HCM  
\*       Check process  
          CASE i\_swnd-process .  
\*\*&---------------------------------------------------------------------\*  
\*\*       Process - Leave request  
\*\*----------------------------------------------------------------------\*  
  
\*            &&\*\*\*\*\*BOC BY UMER EHSAN    5/29/2021  \*\*\*\*\*\*&&  
  
            WHEN c\_process\_lr2.  
  
              CONCATENATE 'Leave Request for employee' initiator\_det-first\_name initiator\_det-last\_name INTO e\_subject SEPARATED BY space.  
              CASE i\_swnd-decision.  
                WHEN c\_decision\_i.  "Initiator  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | Your  { leave\_text } request for the period { startdate } - { enddate }  has been | )  
                  ( | submitted successfully to { approver\_det-first\_name } { approver\_det-last\_name } ( { approver\_det-plstx } ) . | )  
                  ( |  | )  
                 ( | Thanks & Regards, | )  
                 ( |  | )  
                 ( | Department of Human Resources | )  
                  ).  
                WHEN c\_decision\_ar. "Approval required  
                  e\_main\_text = VALUE bcsy\_text(  
                 ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } , | )  
                 ( | | )  
                 ( | { initiator\_det-first\_name   } { initiator\_det-last\_name   } has applied for { leave\_text } request for the period { startdate } - { enddate } | )  
                 ( | Kindly access your "My Inbox" Fiori Tile and take action on leave request by "Recommending or Rejecting" the application.| )  
                 ( |  | )  
                 ( | Additional Information is as under: | )  
                 ( |  | )  
                 ( | Description: { notice\_text } | )  
                 ( | Entitled: { entitled } | )  
                 ( | Availed:  { availed } | )  
                 ( | Balance:   { balance } | )  
                 ( | Current Leave Applied for: { total\_days } | )  
                 ( |  | )  
                 ( | Thanks & Regards, | )  
                 ( |  | )  
                 ( | Department of Human Resources | )  
                 ).  
                WHEN c\_decision\_a.  "Approved  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | Your  { leave\_text } request for the period { startdate } - { enddate }  has been | )  
                  ( | recommended & forwarded by { approver\_det-first\_name } { approver\_det-last\_name } ( { approver\_det-plstx } ) . | )  
                  ( |  | )  
                  ( | Remarks: { decision\_note }.| )  
                  ( |  | )  
                  ( | Thanks & Regards, | )  
                  ( |  | )  
                  ( | Department of Human Resources | )  
                  ).  
  
                WHEN c\_decision\_r.  "Reject  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | Your  { leave\_text } request for the period { startdate } - { enddate }  has been | )  
                  ( | rejected by { approver\_det-first\_name } { approver\_det-last\_name } ( { approver\_det-plstx } ) . | )  
                  ( |  | )  
                  ( | Reason for rejected: { decision\_note } .| )  
                  ( |  | )  
                  ( | Thanks & Regards, | )  
                  ( |  | )  
                  ( | Department of Human Resources | )  
                  ).  
  
                WHEN c\_decision\_f. "Final  
  
  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | { leave\_text } leave request applied by { initiator\_det-first\_name   } { initiator\_det-last\_name   } for the peried { startdate } - { enddate } | )  
                  ( | has been approved by { approver\_det-first\_name } { approver\_det-last\_name } ( { approver\_det-plstx } ).| )  
                  ( |  | )  
                  ( | Remarks: { decision\_note }.| )  
                  ( |  | )  
                  ( | Thanks & Regards, | )  
                  ( |  | )  
                  ( | Department of Human Resources | )  
                  ).  
  
  
              ENDCASE.  
  
\*            &&\*\*\*\*\*BOC BY UMER EHSAN    5/29/2021  \*\*\*\*\*\*&&  
  
  
            WHEN c\_process\_lr.                  "Leave request  
  
  
              CONCATENATE 'Leave Request for employee' initiator\_det-first\_name initiator\_det-last\_name INTO e\_subject SEPARATED BY space.  
              CASE i\_swnd-decision.  
                WHEN c\_decision\_i.  "Initiator  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | Your  { leave\_text } request for the period { startdate } - { enddate }  has been | )  
                  ( | submitted successfully to { approver\_det-first\_name } { approver\_det-last\_name } ( { approver\_det-plstx } ). | )  
                  ( |  | )  
                 ( | Thanks & Regards, | )  
                 ( |  | )  
                 ( | Department of Human Resources | )  
                  ).  
                WHEN c\_decision\_ar. "Approval required  
                  e\_main\_text = VALUE bcsy\_text(  
                 ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } , | )  
                 ( | | )  
                 ( | { initiator\_det-first\_name   } { initiator\_det-last\_name   } has applied for { leave\_text } request for the period { startdate } - { enddate } . | )  
                 ( | Kindly access your "My Inbox" Fiori Tile and take action on leave request by "Accepting or Rejecting" the application.| )  
                 ( |  | )  
                 ( | Additional Information is as under: | )  
                 ( |  | )  
                 ( | Description: { notice\_text } | )  
                 ( | Entitled: { entitled } | )  
                 ( | Availed:  { availed } | )  
                 ( | Balance:   { balance } | )  
                 ( | Current Leave Applied for: { total\_days } | )  
                 ( |  | )  
                 ( | Thanks & Regards, | )  
                 ( |  | )  
                 ( | Department of Human Resources | )  
                 ).  
                WHEN c\_decision\_a.  "Approved  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | Your  { leave\_text } request for the period { startdate } - { enddate }  has been | )  
                  ( | recommended & forwarded by { approver\_det-first\_name } { approver\_det-last\_name } ( { approver\_det-plstx } ). | )  
                  ( |  | )  
                  ( | Remarks: { decision\_note }.| )  
                  ( |  | )  
                  ( | Thanks & Regards, | )  
                  ( |  | )  
                  ( | Department of Human Resources | )  
                  ).  
  
                WHEN c\_decision\_r.  "Reject  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | Your  { leave\_text } request for the period { startdate } - { enddate }  has been | )  
                  ( | rejected by { approver\_det-first\_name } { approver\_det-last\_name } ( { approver\_det-plstx } ). | )  
                  ( |  | )  
                  ( | Reason for rejected: { decision\_note } .| )  
                  ( |  | )  
                  ( | Thanks & Regards, | )  
                  ( |  | )  
                  ( | Department of Human Resources | )  
                  ).  
  
                WHEN c\_decision\_f. "Final  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | { leave\_text } request applied by { initiator\_det-first\_name   } { initiator\_det-last\_name   } for the peried { startdate } - { enddate } | )  
                  ( |  has been approved by { approver\_det-first\_name } { approver\_det-last\_name } ( { approver\_det-plstx } ). | )  
                  ( |  | )  
                  ( | Remarks: { decision\_note }.| )  
                  ( |  | )  
                  ( | Thanks & Regards, | )  
                  ( |  | )  
                  ( | Department of Human Resources | )  
                  ).  
  
  
              ENDCASE.  
\*\*&---------------------------------------------------------------------\*  
\*\*       Process - Travel Request  
\*\*----------------------------------------------------------------------\*  
            WHEN c\_process\_tr.  
              DATA(lv\_pernr) = CONV pernr\_d( instid+0(8) ).  
              DATA(lv\_tripno) = CONV reinr( instid+8(10) ).  
\*           Get trip detail  
              SELECT SINGLE \*  
                FROM ptrv\_head  
                INTO @DATA(ptrv\_head)  
                WHERE pernr EQ @lv\_pernr  
                  AND reinr EQ @lv\_tripno.  
  
\*          Call function to convert start date  
              CALL FUNCTION 'CONVERT\_DATE\_TO\_EXTERNAL'  
                EXPORTING  
                  date\_internal            = ptrv\_head-datv1  
                IMPORTING  
                  date\_external            = startdate  
                EXCEPTIONS  
                  date\_internal\_is\_invalid = 1  
                  OTHERS                   = 2.  
\*          Call function to convert end date  
              CALL FUNCTION 'CONVERT\_DATE\_TO\_EXTERNAL'  
                EXPORTING  
                  date\_internal            = ptrv\_head-datb1  
                IMPORTING  
                  date\_external            = enddate  
                EXCEPTIONS  
                  date\_internal\_is\_invalid = 1  
                  OTHERS                   = 2.  
  
              CASE i\_swnd-decision.  
                WHEN c\_decision\_i.  "Initiator  
                  CONCATENATE 'TR for employee' initiator\_det-first\_name  initiator\_det-last\_name INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                     ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name }, | )  
                     ( | | )  
                     ( | Your Request for Travel Booking (RTB) for the period { startdate } to { enddate } has been  | )  
                     ( | forwarded to your line Manager for Approval.  | )  
                     ( | | )  
                     ( | | )  
                     ( | HR | )  
                      ).  
                WHEN c\_decision\_ar. "Approval required  
                  CONCATENATE 'TR Approval for employee' initiator\_det-first\_name  initiator\_det-last\_name INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   }, | )  
                  ( | | )  
                  ( | Request for Travel Booking (RTB) applied by  { initiator\_det-first\_name  } { initiator\_det-last\_name } for the period { startdate } to { enddate }.  | )  
                  ( | Please go to My Inbox Fiori Tile and take action on Travel Request.  | )  
                  ( | | )  
                  ( | | )  
                  ( | HR | )  
                  ).  
  
                WHEN c\_decision\_a.  "Approved  
                  CONCATENATE 'Travel Request approved for employee' initiator\_det-first\_name  initiator\_det-last\_name INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   }, | )  
                  ( | | )  
                  ( | Your Request for Travel Booking (RTB) for the period  { startdate } TO { enddate } has been approved by { approver\_det-first\_name } { approver\_det-last\_name }. | )  
                  ( | | )  
                  ( | | )  
                  ( | HR | )  
                  ).  
                WHEN c\_decision\_alm."Approved by line manager  
                  CONCATENATE 'TR approved for employee' initiator\_det-first\_name  initiator\_det-last\_name INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   }, | )  
                  ( | | )  
                  ( | Your Request for Travel Booking (RTB) for the period  { startdate } TO { enddate } has been  | )  
                  ( | approved by your line Manager now it will be forwarded for further Approval.| )  
                  ( | | )  
                  ( | | )  
                  ( | HR | )  
                  ).  
                WHEN c\_decision\_rlm."Rejected by line manager  
                  CONCATENATE 'TR rejected for emp.' initiator\_det-first\_name  initiator\_det-last\_name INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   }, | )  
                  ( | | )  
                  ( | Your Request for Travel Booking (RTB) for the period  { startdate } TO { enddate } has been   | )  
                  ( | rejected by your Line Manager.| )  
                  ( | | )  
                  ( | | )  
                  ).  
                  IF NOT decision\_note IS INITIAL.  
                    e\_main\_text = VALUE bcsy\_text( BASE e\_main\_text  
                    ( | Remarks: { decision\_note }.| )  
                    ).  
                    e\_main\_text = VALUE bcsy\_text( BASE e\_main\_text  
                    ( | | )  
                    ( | | )  
                    ( | HR | )  
                    ).  
                  ENDIF.  
  
                WHEN c\_decision\_r.  "Reject  
                  CONCATENATE 'TR rejected for employee' initiator\_det-first\_name  initiator\_det-last\_name INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   }, | )  
                  ( | | )  
                  ( | Your Request for Travel Booking (RTB) for the period  { startdate } to { enddate } has been rejected.| )  
                  ( | | )  
                  ( | | )  
                  ).  
  
                  IF NOT decision\_note IS INITIAL.  
                    e\_main\_text = VALUE bcsy\_text( BASE e\_main\_text  
                    ( | Remarks: { decision\_note }.| )  
                    ).  
                    e\_main\_text = VALUE bcsy\_text( BASE e\_main\_text  
                    ( | | )  
                    ( | | )  
                    ( | HR | )  
                    ).  
                  ENDIF.  
  
  
                WHEN c\_decision\_sbc.  "Travel request for correction  
                  e\_subject = 'TR for correction'.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   }, | )  
                  ( | | )  
                  ( | Your Request for Travel Booking (RTB) for the period  { startdate } TO { enddate } has been sent back for correction| )  
                  ( | by { approver\_det-first\_name } { approver\_det-last\_name }. Please open your Travel Request and take necessary action. | )  
                  ).  
  
                  IF NOT decision\_note IS INITIAL.  
                    e\_main\_text = VALUE bcsy\_text( BASE e\_main\_text  
                    ( | Remarks: { decision\_note }.| )  
                    ).  
                    e\_main\_text = VALUE bcsy\_text( BASE e\_main\_text  
                    ( | | )  
                    ( | | )  
                    ( | HR | )  
                    ).  
                  ENDIF.  
  
                WHEN c\_decision\_fi.  "Final intiator  
                  CONCATENATE 'TR approved for employee' initiator\_det-first\_name  initiator\_det-last\_name INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   }, | )  
                  ( | | )  
                  ( | Your Request for Travel Booking (RTB) for the period  { startdate } TO { enddate } has been approved.  | )  
                  ( | | )  
                  ( | | )  
                  ( | HR | )  
                  ).  
                WHEN c\_decision\_f.  "Final  
                  CONCATENATE 'TR approved for employee' initiator\_det-first\_name  initiator\_det-last\_name INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   }, | )  
                  ( | | )  
                  ( | Request for Travel Booking (RTB) applied by { initiator\_det-first\_name } { initiator\_det-last\_name } for  the period  { startdate } TO { enddate } has been approved.  | )  
                  ( | | )  
                  ( | | )  
                  ( | HR | )  
                  ).  
              ENDCASE.  
  
  
\*\*&---------------------------------------------------------------------\*  
\*\*       Process - Travel Expense  
\*\*----------------------------------------------------------------------\*  
            WHEN c\_process\_te.  
  
              lv\_pernr = instid+0(8) .  
              lv\_tripno = instid+8(10) .  
\*            Get trip detail  
              SELECT SINGLE \*  
              FROM ptrv\_head  
              INTO ptrv\_head  
               WHERE pernr EQ lv\_pernr  
                 AND reinr EQ lv\_tripno.  
  
              SELECT \* FROM  
              ptrv\_srec  
              INTO TABLE @DATA(lt\_ptrv\_srec)  
                    WHERE pernr EQ @lv\_pernr  
                    AND reinr EQ @lv\_tripno.  
  
\*             Trip Statistics - Trip Amounts  
              SELECT SINGLE \*  
                FROM ptrv\_shdr  
                INTO @DATA(ptrv\_shdr)  
                WHERE pernr EQ @lv\_pernr  
                  AND reinr EQ @lv\_tripno.  
              DATA te\_amount   TYPE gesba.  
              DATA te\_curr\_key TYPE waers.  
              LOOP AT lt\_ptrv\_srec INTO DATA(ls\_ptrv\_srec).  
                te\_amount = te\_amount + ls\_ptrv\_srec-loc\_amount.  
                te\_curr\_key = ls\_ptrv\_srec-loc\_curr.  
              ENDLOOP.  
\*              Mileage  
              te\_amount = te\_amount + ptrv\_shdr-pd\_mileage.  
\*             Meals per Diem  
              te\_amount = te\_amount + ptrv\_shdr-pd\_food.  
\*             Accommodation per Diem  
              te\_amount = te\_amount + ptrv\_shdr-pd\_housing.  
              IF te\_curr\_key IS INITIAL.  
                te\_curr\_key = ptrv\_shdr-currency.  
              ENDIF.  
  
\*          Call function to convert start date  
              CALL FUNCTION 'CONVERT\_DATE\_TO\_EXTERNAL'  
                EXPORTING  
                  date\_internal            = ptrv\_head-datv1  
                IMPORTING  
                  date\_external            = startdate  
                EXCEPTIONS  
                  date\_internal\_is\_invalid = 1  
                  OTHERS                   = 2.  
\*           Call function to convert end date  
              CALL FUNCTION 'CONVERT\_DATE\_TO\_EXTERNAL'  
                EXPORTING  
                  date\_internal            = ptrv\_head-datb1  
                IMPORTING  
                  date\_external            = enddate  
                EXCEPTIONS  
                  date\_internal\_is\_invalid = 1  
                  OTHERS                   = 2.  
  
              CASE i\_swnd-decision.  
  
                WHEN c\_decision\_i.  "Initiator  
                  CONCATENATE 'TE for employee' initiator\_det-first\_name  initiator\_det-last\_name INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | Your Travel expense statement of amount  { te\_amount } { te\_curr\_key } has been Submitted for approval. | )  
                  ( | | )  
                  ( |HR | )  
                  ).  
  
                WHEN c\_decision\_ar. "Approval required  
                  CONCATENATE 'TE Approval for employee' initiator\_det-first\_name  initiator\_det-last\_name INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
              ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } , | )  
              ( | | )  
              ( | Travel expense statement of amount { te\_amount } { te\_curr\_key } applied by { initiator\_det-first\_name   } { initiator\_det-last\_name   }. | )  
              ( | Please go to My Inbox Fiori app and tak action on Travel Expense Statment.| )  
              ( |  | )  
              ( | HR | )  
              ).  
  
                WHEN c\_decision\_a.  "Approved  
                  CONCATENATE 'TE approved for employee' initiator\_det-first\_name  initiator\_det-last\_name INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | Your travel expense statement of amount { te\_amount } { te\_curr\_key } | )  
                  ( | has been approved by { approver\_det-first\_name } { approver\_det-last\_name } . | )  
                  ( | | )  
                  ( |HR | )  
                  ).  
  
                WHEN c\_decision\_r.  "Reject  
                  CONCATENATE 'TE rejected for employee' initiator\_det-first\_name  initiator\_det-last\_name INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | Your travel expense statement of amount { te\_amount } { te\_curr\_key } | )  
                  ( | has been rejected by  { approver\_det-first\_name } { approver\_det-last\_name } . | )  
                  ( | | )  
                  ).  
                  IF NOT decision\_note IS INITIAL.  
                    e\_main\_text = VALUE bcsy\_text( BASE e\_main\_text  
                    ( | Remarks: { decision\_note }.| )  
                    ).  
                    e\_main\_text = VALUE bcsy\_text( BASE e\_main\_text  
                    ( | | )  
                    ( | | )  
                    ( | HR | )  
                    ).  
                  ENDIF.  
                WHEN c\_decision\_sbc.  "Travel expense for correction  
                  e\_subject = 'TE for correction'.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   }, | )  
                  ( | | )  
                  ( | Your travel expense statement(TES) has been sent back for correction| )  
                  ( | by { approver\_det-first\_name } { approver\_det-last\_name }. Please open your travel expense statement and take necessary action. | )  
                  ).  
                  IF NOT decision\_note IS INITIAL.  
                    e\_main\_text = VALUE bcsy\_text( BASE e\_main\_text  
                    ( | Remarks: { decision\_note }.| )  
                    ).  
                  ENDIF.  
                  e\_main\_text = VALUE bcsy\_text( BASE e\_main\_text  
                  ( | | )  
                  ( | | )  
                  ( | HR | )  
                  ).  
              ENDCASE.  
  
          ENDCASE.  
\*\*&---------------------------------------------------------------------\*  
\*\*       Module MM  
\*\*----------------------------------------------------------------------\*  
        WHEN c\_module\_mm.  
\*         Check process  
          CASE i\_swnd-process .  
\*\*&---------------------------------------------------------------------\*  
\*\*       Process - MIRO  
\*\*----------------------------------------------------------------------\*  
            WHEN c\_process\_mir.  "MIR7  
              DATA object\_key     TYPE awkey.  
              DATA(document\_number) = CONV belnr\_d( instid+0(10) ).  
              DATA(fiscal\_year)     = CONV gjahr( instid+10(14) ).  
              CONCATENATE document\_number fiscal\_year INTO object\_key.  
\*             Get document number  
              SELECT SINGLE belnr, bukrs, usnam  
              FROM vbkpf  
              INTO ( @DATA(belnr) , @DATA(company\_code) , @DATA(usnam) )  
              WHERE awkey EQ @object\_key.  
  
              SELECT SINGLE \*  
              FROM rseg INTO @DATA(ls\_rseg)  
              WHERE belnr = @document\_number.  
  
              DATA : lv\_name             TYPE thead-tdname,  
                     lv\_header           TYPE thead,  
                     lv\_old\_line\_counter TYPE thead-tdtxtlines,  
                     lt\_lines            TYPE TABLE OF tline,  
                     lv\_lines            TYPE char255,  
\*                     lv\_id               TYPE thead-tdid   VALUE 'F02',  
\*                     lv\_object           TYPE thead-tdname VALUE 'EKKO',  
                     archive\_handle      LIKE  sy-tabix.  
  
  
                 lv\_name = ls\_rseg-ebeln.  
              CALL FUNCTION 'READ\_TEXT'  
                EXPORTING  
\*                  client                  = sy-mandt  
                  id                      = 'F02'  
                  language                = 'E'  
                  name                    = lv\_name  
                  object                  = 'EKKO'  
                  archive\_handle          = archive\_handle  
                IMPORTING  
                  header                  = lv\_header  
                  old\_line\_counter        = lv\_old\_line\_counter  
                TABLES  
                  lines                   = lt\_lines  
                EXCEPTIONS  
                  id                      = 1  
                  language                = 2  
                  name                    = 3  
                  not\_found               = 4  
                  object                  = 5  
                  reference\_check         = 6  
                  wrong\_access\_to\_archive = 7  
                  OTHERS                  = 8.  
              IF sy-subrc <> 0.  
\* Implement suitable error handling here  
              ENDIF.  
  
              LOOP AT lt\_lines INTO DATA(ls\_lines).  
                DATA(line) = ls\_lines-tdline.  
\*                lv\_lines   = lv\_lines && lv\_lines.  
  
                CONCATENATE line lv\_lines INTO lv\_lines SEPARATED BY space.  
              ENDLOOP.  
  
  
  
              CASE i\_swnd-decision.  
                WHEN c\_decision\_i.  "Initiator  
                  CONCATENATE 'Invoice' document\_number INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } . | )  
                  ( | The Invoice no { document\_number } , Company code { company\_code } and Fiscal Year { fiscal\_year } | )  
                  ( | has been forwarded for approval. | )  
                  ( | | )  
                  ( | | )  
                  ( | Regards,   | )  
                  ( | NDRMF Notification  | )  
  
                  ).  
  
                WHEN c\_decision\_ar. "Approval required  
                  CONCATENATE 'Release invoice ' document\_number INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } . | )  
                  ( | The Invoice no { document\_number } , Company code { company\_code } and Fiscal Year { fiscal\_year } | )  
                  ( | Parked by { initiator\_det-first\_name  }  { initiator\_det-last\_name } please go to My Inbox Fiori App and take | )  
                  ( | further action to process the document further i.e. approve/reject | )  
                  ( | | )  
                  ( | | )  
                  ( | Regards,   | )  
                  ( | NDRMF Notification  | )  
  
                  ).  
                WHEN c\_decision\_a.  "Approved  
                  CONCATENATE 'Approved Invoice' document\_number INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } . | )  
                  ( | The Invoice no { document\_number } , Company code { company\_code } and Fiscal Year { fiscal\_year } | )  
                  ( | has been approved by { approver\_det-first\_name } { approver\_det-last\_name } | )  
                  ( | | )  
                  ( | | )  
                  ( | Regards,   | )  
                  ( | NDRMF Notification  | )  
  
                  ).  
                WHEN c\_decision\_r.  "Reject  
                  CONCATENATE 'Rejected Invoice' document\_number INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } . | )  
                  ( | The Invoice no { document\_number } , Company code { company\_code } and Fiscal Year { fiscal\_year } | )  
                  ( | has been rejected { approver\_det-first\_name } { approver\_det-last\_name } | )  
                  ( | | )  
                  ( | | )  
                  ( | Regards,   | )  
                  ( | NDRMF Notification  | )  
  
                  ).  
\*                WHEN C\_DECISION\_F. "Final  
\*                  CONCATENATE 'Fully approved Invoice' DOCUMENT\_NUMBER INTO E\_SUBJECT SEPARATED BY SPACE.  
\*                  E\_MAIN\_TEXT = VALUE BCSY\_TEXT(  
\*                  ( | Dear  { I\_USRD-FIRST\_NAME  } { I\_USRD-LAST\_NAME  } . | )  
\*                  ( | The Invoice no { DOCUMENT\_NUMBER } , Company code { COMPANY\_CODE } and Fiscal Year { FISCAL\_YEAR } | )  
\*                  ( | has been fully approved  | )  
\*                  ( | | )  
\*                  ( | | )  
\*                  ( | Regards,   | )  
\*                  ( | NDRMF Notification  | )  
                WHEN c\_decision\_f. "Final  
                  CONCATENATE 'Fully approved Invoice' document\_number INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } . | )  
                  ( | INVOICE DOCUMENT NO.{ document\_number } | )  
                  ( | SUBJECT.{ lv\_lines } | )  
                  ( | HAS BEEN FULLY APPROVED BY APPROVING AUTHORITY.  | )  
                  ( | | )  
                  ( | | )  
                  ( | Regards,   | )  
                  ( | NDRMF Notification  | )  
                  ).  
  
              ENDCASE.  
\*\*&---------------------------------------------------------------------\*  
\*\*       Process - Purchase order  
\*\*----------------------------------------------------------------------\*  
            WHEN c\_process\_po.  "Purchase order  
              DATA(ebeln) = CONV ekko-ebeln( instid ).  
              CASE i\_swnd-decision.  
                WHEN c\_decision\_ar.  "Approval required  
                  CONCATENATE 'Release purchase Order' ebeln '.' INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } . | )  
                  ( | You have received purchase Order number { ebeln }. | )  
                  ( | Kindly go to 'My Inbox Fiori App' to review, comment & approve for further processing | )  
                  ( | Regards,   | )  
                  ( | NDRMF Notification  | )  
  
                  ).  
                WHEN c\_decision\_a.  "Approved  
                  CONCATENATE 'Purchase Order' ebeln 'Approved.' INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } , | )  
                  ( | Your Purchase order document { ebeln } has been approved by { approver\_det-first\_name } { approver\_det-last\_name }. | )  
                  ( | Regards,   | )  
                  ( | NDRMF Notification  | )  
  
                  ).  
  
                WHEN c\_decision\_r.  "Reject  
                  CONCATENATE 'Purchase Order' ebeln 'Rejected.' INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } , | )  
                  ( | Your Purchase order document { ebeln } has been rejected by { approver\_det-first\_name } { approver\_det-last\_name }. | )  
                  ( | Regards,   | )  
                  ( | NDRMF Notification  | )  
  
                  ).  
  
                WHEN c\_decision\_f. "Final  
                  CONCATENATE 'Purchase Order' ebeln 'Approved.' INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } , | )  
                  ( | Your Purchase order document { ebeln } has been fully approved  | )  
                  ( | Regards,   | )  
                  ( | NDRMF Notification  | )  
                  ).  
  
              ENDCASE.  
\*\*&---------------------------------------------------------------------\*  
\*\*       Process - Purchase requisition  
\*\*----------------------------------------------------------------------\*  
            WHEN c\_process\_pr.  "Purchase requisition  
              ebeln = CONV ekko-ebeln( instid ).  
              CASE i\_swnd-decision.  
                WHEN c\_decision\_ar.  "Approval required  
                  CONCATENATE 'Release purchase requisition' ebeln '.' INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } . | )  
                  ( | You have received purchase requisition number { ebeln }. in your inbox | )  
                  ( | Kindly go to 'My Inbox Fiori App' TO review, comment & approve FOR further processing. | )  
                  ( | Regards,   | )  
                  ( | NDRMF Notification  | )  
                  ).  
                WHEN c\_decision\_a.  "Approved  
                  CONCATENATE 'Purchase requisition' ebeln 'Approved.' INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } , | )  
                  ( | Your Purchase requisition document { ebeln } has been approved by { approver\_det-first\_name } { approver\_det-last\_name }. | )  
                  ( | Regards,   | )  
                  ( | NDRMF Notification  | )  
                  ).  
  
                WHEN c\_decision\_r.  "Reject  
                  CONCATENATE 'Purchase requisition' ebeln 'Rejected.' INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } , | )  
                  ( | Your Purchase requisition document { ebeln } has been rejected by { approver\_det-first\_name } { approver\_det-last\_name }. | )  
                  ( | Regards,   | )  
                  ( | NDRMF Notification  | )  
                  ).  
  
                WHEN c\_decision\_f. "Final  
                  CONCATENATE 'Purchase requisition' ebeln 'Approved.' INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } , | )  
                  ( | Your Purchase requisition document { ebeln } has been fully approved  | )  
                  ( | Regards,   | )  
                  ( | NDRMF Notification  | )  
                  ).  
  
              ENDCASE.  
  
          ENDCASE.  
\*\*&---------------------------------------------------------------------\*  
\*\*       Module FI  
\*\*----------------------------------------------------------------------\*  
        WHEN c\_module\_fi.  
\*         Check process  
          CASE i\_swnd-process.  
\*\*&---------------------------------------------------------------------\*  
\*\*       Process - Parking Posting  
\*\*----------------------------------------------------------------------\*  
            WHEN c\_process\_pp.                     "Parking postin  
              company\_code    = instid+0(4) .  
              document\_number = instid+4(10) .  
              fiscal\_year     = instid+14(4) .  
  
\*             GAccounting Document Header  
              SELECT SINGLE \*  
              FROM bkpf INTO @DATA(ls\_bkpf)  
                    WHERE bukrs = @company\_code AND belnr = @document\_number AND gjahr = @fiscal\_year.  
  
  
              CASE i\_swnd-decision.  
                WHEN c\_decision\_i. "Initator  
                  CONCATENATE 'Document no' document\_number 'sent for review' INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | The Document no { document\_number } company code { company\_code } fiscal year { fiscal\_year } | )  
                  ( | was processed by you.  | )  
                  ( | | )  
                  ( | | )  
                  ( | Regards,  | )  
                  ( | NDRMF Notification.  | )  
                  ).  
                WHEN c\_decision\_ar."Approval required  
                  CONCATENATE 'Release Amount - Document no ' document\_number '.'  INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | The document no { document\_number } company code { company\_code } fiscal year { fiscal\_year } | )  
                  ( | is now available for you to further process. Kindly do the needfull.  | )  
                  ( | | )  
                  ( | | )  
                  ( | Regards,  | )  
                  ( | NDRMF Notification.  | )  
                  ).  
                WHEN c\_decision\_a. "Approved  
                  CONCATENATE 'Document no' document\_number 'approved' INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | Document NO { document\_number } company code { company\_code } fiscal year { fiscal\_year } | )  
                  ( | and reference { ls\_bkpf-xblnr } approved. | )  
                  ).  
                  IF NOT decision\_note IS INITIAL.  
                    e\_main\_text = VALUE bcsy\_text( BASE e\_main\_text  
                    ( | Remarks: { decision\_note }.| )  
                    ).  
                  ENDIF.  
                WHEN c\_decision\_r. "Reject  
                  CONCATENATE 'Document no' document\_number 'rejected' INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name   } { i\_usrd-last\_name   } , | )  
                  ( | | )  
                  ( | The  Document no { document\_number } was rejected by { approver\_det-first\_name } { approver\_det-last\_name } | )  
                  ( | Kindly make necessary adjustments/corrections.    | )  
                  ( | | )  
                  ( | | )  
                  ( | Regards,  | )  
                  ( | NDRMF Notification.  | )  
                  ).  
  
  
\*                WHEN C\_DECISION\_F. "Final  
\*                  CONCATENATE 'Document no' DOCUMENT\_NUMBER 'posted' INTO E\_SUBJECT SEPARATED BY SPACE.  
\*                  E\_MAIN\_TEXT = VALUE BCSY\_TEXT(  
\*                  ( | Dear  { APPROVER\_DET-FIRST\_NAME   } { I\_USRD-LAST\_NAME   } , | )  
\*                  ( | | )  
\*                  ( | The Document no { DOCUMENT\_NUMBER } was posted by you. | )  
\*                  ( | | )  
\*                  ( | | )  
\*                  ( | Regards,  | )  
\*                  ( | NDRMF Notification.  | )  
\*  
\*                  ).  
                WHEN c\_decision\_f. "Final    "BOC By Ehsan TMC - 10.09.2021  
                  CONCATENATE 'Fully approved Invoice' document\_number INTO e\_subject SEPARATED BY space.  
                  e\_main\_text = VALUE bcsy\_text(  
                  ( | Dear  { i\_usrd-first\_name  } { i\_usrd-last\_name  } . | )  
                  ( | PAYMENT DOCUMENT NO.{ document\_number } | )  
                  ( | HAS BEEN APPROVED BY APPROVING AUTHORITY SUBMITTED FOR CHEQUE MARKING PLEASE.  | )  
                  ( | | )  
                  ( | | )  
                  ( | Regards,   | )  
                  ( | NDRMF Notification  | )  
                  ).  
                  "EOC By Ehsan TMC - 10.09.2021  
              ENDCASE.  
          ENDCASE.  
      ENDCASE.  
  
    ENDMETHOD.  
  
  
\* <SIGNATURE>---------------------------------------------------------------------------------------+  
\* | Static Private Method ZCL\_SWF\_NOTI\_UTIL=>GET\_NOTIF\_DET  
\* +-------------------------------------------------------------------------------------------------+  
\* | [--->] I\_SWND                         TYPE        ZST\_SWND(optional)  
\* | [<---] E\_USRD                         TYPE        TT\_USRD  
\* +--------------------------------------------------------------------------------------</SIGNATURE>  
METHOD get\_notif\_det.  
\* Get workflow id  
  SELECT SINGLE top\_wi\_id  
  FROM swwwihead  
  INTO @DATA(top\_wi\_id)  
        WHERE wi\_id EQ @i\_swnd-workitem.  
\*   Get all workfitems of request  
  SELECT \*  
  FROM swwwihead  
  INTO TABLE @DATA(lt\_swwwihead)  
        WHERE top\_wi\_id EQ @top\_wi\_id.  
\*   Get workflow required detail flags  
  SELECT SINGLE \*  
  FROM zswnrd  
  INTO @data(ls\_zswnrd)  
  
        WHERE zmodule  EQ @i\_swnd-module  
        AND process  EQ @i\_swnd-process  
        AND decision EQ @i\_swnd-decision.  
  
  
\*  IF gs\_swnrd-oappr EQ abap\_true. "If other approvers required  
\*\*     Get other approvers  
\*    SELECT \*  
\*    FROM zswnu  
\*    INTO TABLE @DATA(lt\_swnu)  
\*          WHERE zmodule  EQ @i\_swnd-module  
\*          AND   process  EQ @i\_swnd-process  
\*          AND   decision EQ @i\_swnd-decision.  
\*\*     loop to get emails  
\*    e\_usrd = VALUE tt\_usrd( BASE e\_usrd FOR swnu IN lt\_swnu WHERE ( otype = c\_otype\_e )  
\*    ( otype = c\_otype\_e  email =  swnu-email  
\*    first\_name = swnu-first\_name last\_name = swnu-last\_name ) ).  
\*\*     loop to get users  
\*    e\_usrd = VALUE tt\_usrd( BASE e\_usrd FOR swnu IN lt\_swnu WHERE ( otype = c\_otype\_us )  
\*    ( user = CONV usrid( swnu-objid ) )  )  .  
\*\*     loop to get employees  
\*    e\_usrd = VALUE tt\_usrd( BASE e\_usrd FOR swnu IN lt\_swnu WHERE ( otype EQ  c\_otype\_p  )  
\*    (  objid = swnu-objid  otype = swnu-otype pernr = CONV pernr\_d( swnu-objid ) ) ).  
\*\*     loop to get positions  
\*    e\_usrd = VALUE tt\_usrd( BASE e\_usrd FOR swnu IN lt\_swnu WHERE ( otype EQ  c\_otype\_s )  
\*    (  objid = swnu-objid  otype = swnu-otype ) ).  
\*  ENDIF.  
\*  IF gs\_swnrd-all\_aprv EQ abap\_true. "If all approvers required  
\*\*     Loop to get all approvers  
\*    e\_usrd = VALUE tt\_usrd( BASE e\_usrd FOR swwwihead IN lt\_swwwihead WHERE ( wi\_type = c\_wi\_type\_w )  
\*    ( user = CONV sysid( swwwihead-wi\_aagent )  otype = c\_otype\_us )  ).  
\*  ENDIF.  
  
\*  IF gs\_swnrd-vendor EQ abap\_true. "If all approvers required  
\*  
\*\*   Get user master detial  
\*    zcl\_swf\_noti\_util=>hold\_vendor\_det(  
\*    i\_swnd = i\_swnd  
\*    ).  
\*  
\*    e\_usrd = VALUE tt\_usrd( BASE e\_usrd ( vendor\_det ) ).  
\*  
\*  ENDIF.  
  
  DATA(t\_usrd) = e\_usrd.  
\*   Get user master detial  
  zcl\_swf\_noti\_util=>get\_umd(  
  CHANGING  
    c\_usrd = t\_usrd  
    ).  
  
\*   IF ls\_zswnrd-vendor EQ abap\_true. "If all approvers required  
\*  
\*\*   Get user master detial  
\*    zcl\_swf\_noti\_util=>hold\_vendor\_det(  
\*    i\_swnd = i\_swnd  
\*    ).  
\*  
\*    e\_usrd = VALUE tt\_usrd( BASE e\_usrd ( vendor\_det ) ).  
\*  
\*  ENDIF.  
  
  
  IF ls\_zswnrd-initiator EQ abap\_true.  "If initiator required  
    e\_usrd = VALUE tt\_usrd( BASE e\_usrd ( initiator\_det  ) ).  
  
  ENDIF.  
  
  IF ls\_zswnrd-aprv EQ abap\_true. "If approver required  
    e\_usrd = VALUE tt\_usrd( BASE e\_usrd ( approver\_det  ) ).  
  
  ENDIF.  
  
  IF ls\_zswnrd-oappr EQ abap\_true. "If other approvers required  
\*     Get other approvers  
    SELECT \*  
    FROM zswnu  
    INTO TABLE @DATA(lt\_swnu)  
          WHERE zmodule  EQ @i\_swnd-module  
          AND   process  EQ @i\_swnd-process  
          AND   decision EQ @i\_swnd-decision.  
\*     loop to get emails  
    e\_usrd = VALUE tt\_usrd( BASE e\_usrd FOR swnu IN lt\_swnu WHERE ( otype = c\_otype\_e )  
    ( otype = c\_otype\_e  email =  swnu-email  
    first\_name = swnu-first\_name last\_name = swnu-last\_name ) ).  
\*     loop to get users  
    e\_usrd = VALUE tt\_usrd( BASE e\_usrd FOR swnu IN lt\_swnu WHERE ( otype = c\_otype\_us )  
    ( user = CONV usrid( swnu-objid ) )  )  .  
\*     loop to get employees  
    e\_usrd = VALUE tt\_usrd( BASE e\_usrd FOR swnu IN lt\_swnu WHERE ( otype EQ  c\_otype\_p  )  
    (  objid = swnu-objid  otype = swnu-otype pernr = CONV pernr\_d( swnu-objid ) ) ).  
\*     loop to get positions  
    e\_usrd = VALUE tt\_usrd( BASE e\_usrd FOR swnu IN lt\_swnu WHERE ( otype EQ  c\_otype\_s )  
    (  objid = swnu-objid  otype = swnu-otype ) ).  
  ENDIF.  
  
  IF ls\_zswnrd-all\_aprv EQ abap\_true. "If all approvers required  
\*     Loop to get all approvers  
    e\_usrd = VALUE tt\_usrd( BASE e\_usrd FOR swwwihead IN lt\_swwwihead WHERE ( wi\_type = c\_wi\_type\_w )  
    ( user = CONV sysid( swwwihead-wi\_aagent )  otype = c\_otype\_us )  ).  
  ENDIF.  
  
  IF ls\_zswnrd-dnote EQ c\_deci\_nr.  
\*   hold decision note  
    zcl\_swf\_noti\_util=>hold\_dec\_note(  
    i\_swnd = i\_swnd  
    ).  
  ENDIF.  
  
  ENDMETHOD.  
  
  
\* <SIGNATURE>---------------------------------------------------------------------------------------+  
\* | Static Public Method ZCL\_SWF\_NOTI\_UTIL=>GET\_STATUS  
\* +-------------------------------------------------------------------------------------------------+  
\* | [--->] I\_REQSTATUS                    TYPE        TIM\_REQ\_STATUS  
\* | [<---] E\_REQSTATUS                    TYPE        TIM\_REQ\_STATUS  
\* +--------------------------------------------------------------------------------------</SIGNATURE>  
  method GET\_STATUS.  
    COMMIT WORK AND WAIT.  
    E\_REQSTATUS = I\_REQSTATUS.  
  
  endmethod.  
  
  
\* <SIGNATURE>---------------------------------------------------------------------------------------+  
\* | Static Private Method ZCL\_SWF\_NOTI\_UTIL=>GET\_UMD  
\* +-------------------------------------------------------------------------------------------------+  
\* | [<-->] C\_USRD                         TYPE        TT\_USRD(optional)  
\* +--------------------------------------------------------------------------------------</SIGNATURE>  
  METHOD get\_umd.  
    DATA(lt\_emails) = c\_usrd.  
    DELETE lt\_emails     WHERE otype NE c\_otype\_e.  
    IF NOT lt\_emails IS INITIAL..  
\*     Get pernr from emailz  
      SELECT \*  
        FROM pa0105  
        INTO TABLE @DATA(lt\_pa0105)  
        FOR ALL ENTRIES IN @lt\_emails  
        WHERE usrid\_long EQ @lt\_emails-email  
          AND endda      EQ @c\_endda  
          AND subty      EQ @c\_subty\_email.  
      IF sy-subrc EQ 0.  
\*       Loop to modify pernr agains email  
        LOOP AT c\_usrd ASSIGNING FIELD-SYMBOL(<fs\_usrd>) WHERE otype EQ c\_otype\_e.  
          IF line\_exists( lt\_pa0105[ usrid\_long = <fs\_usrd>-email ] ).  
\*           Read table to get pernr  
            DATA(ls\_pa0105) = lt\_pa0105[ usrid\_long = <fs\_usrd>-email ].  
            <fs\_usrd>-pernr = ls\_pa0105-pernr.  
            <fs\_usrd>-objid = ls\_pa0105-pernr.  
            CLEAR ls\_pa0105.  
          ENDIF.  
\*          UNASSIGN <fs\_usrd>.  
        ENDLOOP.  
        CLEAR lt\_pa0105[].  
      ENDIF.  
    ENDIF.  
    DATA(lt\_users) = c\_usrd.  
    DELETE lt\_users      WHERE otype NE c\_otype\_us.  
    IF NOT lt\_users IS INITIAL.  
\*   Get pernrs from users  
      SELECT \*  
      FROM pa0105  
      INTO TABLE lt\_pa0105  
            FOR ALL ENTRIES IN lt\_users  
            WHERE usrid EQ lt\_users-user  
            AND   endda      EQ c\_endda  
            AND   subty      EQ c\_subty\_user.  
      IF sy-subrc EQ 0.  
\*      loop to get pernr from usrid  
        LOOP AT c\_usrd ASSIGNING <fs\_usrd> WHERE otype EQ c\_otype\_us.  
          IF line\_exists( lt\_pa0105[ usrid = <fs\_usrd>-user ] ).  
\*           Read table to get pernr  
            ls\_pa0105 = lt\_pa0105[ usrid = <fs\_usrd>-user ].  
            <fs\_usrd>-pernr = ls\_pa0105-pernr.  
            <fs\_usrd>-objid = ls\_pa0105-pernr.  
            CLEAR ls\_pa0105.  
          ENDIF.  
\*          UNASSIGN <fs\_usrd>.  
        ENDLOOP.  
        CLEAR lt\_pa0105[].  
      ENDIF.  
    ENDIF.  
    DATA(lt\_employees) = c\_usrd.  
    DELETE lt\_employees WHERE otype NE c\_otype\_e AND otype NE c\_otype\_p.  
    IF NOT lt\_employees IS INITIAL.  
\*     Get users of employee  
      SELECT \*  
      FROM pa0105  
      INTO TABLE lt\_pa0105  
      FOR ALL ENTRIES IN lt\_employees  
      WHERE pernr EQ lt\_employees-pernr  
        AND endda EQ c\_endda  
        AND subty EQ c\_subty\_user.  
      IF sy-subrc EQ 0.  
\*       loop to get pernr from usrid  
        LOOP AT c\_usrd ASSIGNING <fs\_usrd> WHERE otype EQ c\_otype\_e AND otype EQ c\_otype\_p.  
          IF line\_exists( lt\_pa0105[ pernr = <fs\_usrd>-pernr ] ).  
\*         Read table to get pernr  
            ls\_pa0105 = lt\_pa0105[ pernr = <fs\_usrd>-pernr ].  
            <fs\_usrd>-user = ls\_pa0105-usrid.  
            CLEAR ls\_pa0105.  
          ENDIF.  
\*          UNASSIGN <fs\_usrd>.  
        ENDLOOP.  
        CLEAR lt\_pa0105[].  
      ENDIF.  
    ENDIF.  
    CLEAR lt\_employees[].  
    lt\_employees = c\_usrd.  
    DELETE lt\_employees WHERE otype NE c\_otype\_us AND otype NE c\_otype\_p.  
    IF NOT lt\_employees IS INITIAL.  
\*     Get emails of employees  
      SELECT \*  
      FROM pa0105  
      INTO TABLE lt\_pa0105  
      FOR ALL ENTRIES IN lt\_employees  
      WHERE pernr EQ lt\_employees-pernr  
      AND endda EQ c\_endda  
      AND subty EQ c\_subty\_email.  
      IF sy-subrc EQ 0.  
\*       loop to get emails from employee  
        LOOP AT c\_usrd ASSIGNING <fs\_usrd> WHERE otype EQ c\_otype\_us or otype EQ c\_otype\_p.  
          IF line\_exists( lt\_pa0105[ pernr = <fs\_usrd>-pernr ] ).  
\*           Read table to get pernr  
            ls\_pa0105 = lt\_pa0105[ pernr = <fs\_usrd>-pernr ].  
            <fs\_usrd>-email = ls\_pa0105-usrid\_long.  
            CLEAR ls\_pa0105.  
          ENDIF.  
\*          UNASSIGN <fs\_usrd>.  
        ENDLOOP.  
        CLEAR lt\_pa0105[].  
      ENDIF.  
    ENDIF.  
    IF NOT c\_usrd IS INITIAL.  
\*     Get employee names  
      SELECT \*  
        FROM pa0002  
        INTO TABLE @DATA(lt\_pa0002)  
        FOR ALL ENTRIES IN @c\_usrd  
        WHERE pernr EQ @c\_usrd-pernr  
          AND endda EQ @c\_endda.  
      IF sy-subrc EQ 0.  
        LOOP AT c\_usrd ASSIGNING <fs\_usrd>.  
          IF line\_exists( lt\_pa0002[ pernr = <fs\_usrd>-pernr ] ).  
            IF <fs\_usrd>-otype NE c\_otype\_e. "added 07-07-2020 "User detail not required if only emails are maintained in other apprvers tables  
              DATA(ls\_pa0002) = lt\_pa0002[ pernr = <fs\_usrd>-pernr ].  
              <fs\_usrd>-first\_name = ls\_pa0002-vorna.  
              <fs\_usrd>-second\_name = ls\_pa0002-nach2.  
              <fs\_usrd>-last\_name =   ls\_pa0002-nachn.  
              <fs\_usrd>-full\_name =   ls\_pa0002-cname.  
              CASE <fs\_usrd>-wf\_aty.  
                WHEN c\_wf\_aty\_i.  
                  initiator\_det = <fs\_usrd>.  
                WHEN c\_wf\_aty\_a.  
                  approver\_det = <fs\_usrd>.  
              ENDCASE.  
              CLEAR ls\_pa0002.  
            ENDIF.  
          ENDIF.  
\*          UNASSIGN <fs\_usrd>.  
        ENDLOOP.  
      ENDIF.  
    ENDIF.  
  ENDMETHOD.  
  
  
\* <SIGNATURE>---------------------------------------------------------------------------------------+  
\* | Static Private Method ZCL\_SWF\_NOTI\_UTIL=>HOLD\_AGENTS  
\* +-------------------------------------------------------------------------------------------------+  
\* | [--->] I\_SWND                         TYPE        ZST\_SWND(optional)  
\* | [--->] I\_AGENTS                       TYPE        ZTT\_AGENTS(optional)  
\* +--------------------------------------------------------------------------------------</SIGNATURE>  
  METHOD hold\_agents.  
\*   append initiator  
    DATA(lt\_agents) = VALUE tt\_usrd( ( user = CONV sysid( i\_swnd-initiator+2 ) otype = c\_otype\_us wf\_aty = c\_wf\_aty\_i )  ).  
\*   Append approver  
    IF NOT i\_swnd-approver IS INITIAL.  
      lt\_agents = VALUE tt\_usrd( BASE lt\_agents ( user = CONV sysid( i\_swnd-approver+2 ) otype = c\_otype\_us wf\_aty = c\_wf\_aty\_a ) ).  
    ELSE.  
\*     Get workflow id  
      SELECT SINGLE top\_wi\_id  
      FROM swwwihead  
      INTO @DATA(top\_wi\_id)  
            WHERE wi\_id EQ @i\_swnd-workitem.  
\*     Get all workfitems of request  
      SELECT \*  
      FROM swwwihead  
      INTO TABLE @DATA(lt\_swwwihead)  
            WHERE top\_wi\_id EQ @top\_wi\_id  
              AND  wi\_type = @c\_wi\_type\_w.  
      SORT lt\_swwwihead DESCENDING BY wi\_cd wi\_ct.  
      CASE i\_swnd-decision.  
        WHEN c\_decision\_ar or c\_decision\_i.  
          if line\_exists( i\_agents[ 1 ] ).  
            data(ls\_agent) = i\_agents[ 1 ].  
              lt\_agents = VALUE tt\_usrd( BASE lt\_agents ( user = CONV sysid( ls\_agent+2 ) otype = c\_otype\_us wf\_aty = c\_wf\_aty\_a ) ).  
              clear ls\_agent.  
          endif.  
  
\*          IF line\_exists( lt\_swwwihead[ wi\_stat = c\_wi\_stat\_ready ] ).  
\*            DATA(ls\_swwwihead) = lt\_swwwihead[ wi\_stat = c\_wi\_stat\_ready ].  
\*            SELECT SINGLE \*  
\*             FROM swwuserwi  
\*             INTO @DATA(ls\_swwuserwi)  
\*              WHERE wi\_id EQ @ls\_swwwihead-wi\_id.  
\*            IF NOT ls\_swwuserwi-user\_id IS INITIAL.  
\*            ENDIF.  
\*          ENDIF.  
        WHEN OTHERS.  
          IF line\_exists( lt\_swwwihead[ wi\_stat = c\_wi\_stat\_completed ] ).  
            data(ls\_swwwihead) = lt\_swwwihead[ wi\_stat = c\_wi\_stat\_completed ].  
            lt\_agents = VALUE tt\_usrd( BASE lt\_agents ( user = CONV sysid( ls\_swwwihead-wi\_aagent ) otype = c\_otype\_us wf\_aty = c\_wf\_aty\_a ) ).  
          ENDIF.  
      ENDCASE.  
    ENDIF.  
    CHECK NOT lt\_agents IS INITIAL.  
\*   Get pernrs from users  
    SELECT \*  
    FROM pa0105  
    INTO TABLE @DATA(lt\_employees)  
    FOR ALL ENTRIES IN @lt\_agents  
     WHERE usrid EQ @lt\_agents-user  
     AND   endda EQ @c\_endda  
     AND   subty EQ @c\_subty\_user.  
  
    CHECK NOT lt\_employees IS INITIAL.  
\*   Get employee names  
    SELECT \*  
    FROM pa0002  
    INTO TABLE @DATA(lt\_pa0002)  
    FOR ALL ENTRIES IN @lt\_employees  
    WHERE pernr EQ @lt\_employees-pernr  
      AND endda EQ @c\_endda.  
  
\*     get positions  
  
       SELECT \*  
    FROM pa0001  
    INTO TABLE @DATA(lt\_pa0001)  
    FOR ALL ENTRIES IN @lt\_employees  
    WHERE pernr EQ @lt\_employees-pernr  
      AND endda EQ @c\_endda.  
  
\*   GET POSITION TEXT ADDED BY UMER EHSAN 30/5/2021  
  
    SELECT \*  
    FROM HRP1000  
    INTO TABLE @DATA(LT\_HRP1000)  
    FOR ALL ENTRIES IN @lt\_pa0001  
    WHERE objid EQ @lt\_pa0001-plans  
      AND endda EQ @c\_endda  
      AND OTYPE EQ 'S'  
      AND PLVAR EQ '01'.  
  
\*   Get email addresss  
    SELECT \*  
    FROM pa0105  
    INTO TABLE @DATA(lt\_emails)  
    FOR ALL ENTRIES IN @lt\_employees  
    WHERE pernr EQ @lt\_employees-pernr  
      AND endda EQ @c\_endda  
      AND subty EQ @c\_subty\_email.  
  
    LOOP AT lt\_agents ASSIGNING FIELD-SYMBOL(<fs\_agent>).  
\*     Check employees  
      IF line\_exists( lt\_employees[ usrid = <fs\_agent>-user ] ).  
\*     Read table to get pernr  
        DATA(ls\_employee) = lt\_employees[ usrid = <fs\_agent>-user ].  
        <fs\_agent>-pernr = ls\_employee-pernr.  
        <fs\_agent>-objid = ls\_employee-pernr.  
        CLEAR ls\_employee.  
\*      Check employee names  
        IF line\_exists( lt\_pa0002[ pernr = <fs\_agent>-pernr ] ).  
\*       Read table to get employee names  
          DATA(ls\_pa0002) = lt\_pa0002[ pernr = <fs\_agent>-pernr ].  
          <fs\_agent>-first\_name = ls\_pa0002-vorna.  
          <fs\_agent>-second\_name = ls\_pa0002-nach2.  
          <fs\_agent>-last\_name =   ls\_pa0002-nachn.  
          <fs\_agent>-full\_name =   ls\_pa0002-cname.  
          CLEAR ls\_pa0002.  
        ENDIF.  
  
\*        get employee postions  
        IF line\_exists( lt\_pa0001[ pernr = <fs\_agent>-pernr ] ).  
\*       Read table to get employee postions  
          DATA(ls\_pa0001) = lt\_pa0001[ pernr = <fs\_agent>-pernr ].  
          <fs\_agent>-PLANS = ls\_pa0001-PLANS.  
          CLEAR ls\_pa0001.  
        ENDIF.  
  
\*       Check designation text  added by umer ehsan 30/05/2021  
  
        IF line\_exists( lt\_hrp1000[ objid = <fs\_agent>-plans ] ).  
\*       Read table to get employee names  
          DATA(ls\_hrp1000) = lt\_hrp1000[ objid = <fs\_agent>-plans ].  
          <fs\_agent>-PLANS = ls\_hrp1000-OBJID.  
          <fs\_agent>-PLSTX = ls\_hrp1000-STEXT.  
          CLEAR ls\_hrp1000.  
        ENDIF.  
\*       Check emails  
        IF line\_exists( lt\_emails[ pernr = <fs\_agent>-pernr ] ).  
\*       Read table to get emails  
          DATA(ls\_email)  = lt\_emails[ pernr = <fs\_agent>-pernr ].  
          <fs\_agent>-email = ls\_email-usrid\_long.  
          CLEAR ls\_email.  
        ENDIF.  
        CASE <fs\_agent>-wf\_aty.  
          WHEN  c\_wf\_aty\_i.  
            initiator\_det = <fs\_agent>.  
          WHEN  c\_wf\_aty\_a.  
            approver\_det = <fs\_agent>.  
        ENDCASE.  
      ENDIF.  
    ENDLOOP.  
  ENDMETHOD.  
  
  
\* <SIGNATURE>---------------------------------------------------------------------------------------+  
\* | Static Private Method ZCL\_SWF\_NOTI\_UTIL=>HOLD\_DEC\_NOTE  
\* +-------------------------------------------------------------------------------------------------+  
\* | [--->] I\_SWND                         TYPE        ZST\_SWND(optional)  
\* +--------------------------------------------------------------------------------------</SIGNATURE>  
  METHOD hold\_dec\_note.  
  
    DATA :simple\_container         TYPE STANDARD TABLE OF swr\_cont,  
          message\_lines            TYPE STANDARD TABLE OF swr\_messag,  
          message\_struct           TYPE STANDARD TABLE OF swr\_mstruc,  
          subcontainer\_bor\_objects TYPE STANDARD TABLE OF swr\_cont,  
          lt\_objects               TYPE STANDARD TABLE OF swr\_cont,  
          lt\_content               TYPE TABLE OF solisti1.  
\*   Read workflow container  
    CALL FUNCTION 'SAP\_WAPI\_READ\_CONTAINER'  
      EXPORTING  
        workitem\_id              = i\_swnd-workitem  
      TABLES  
        simple\_container         = simple\_container  
        message\_lines            = message\_lines  
        message\_struct           = message\_struct  
        subcontainer\_bor\_objects = subcontainer\_bor\_objects  
        subcontainer\_all\_objects = lt\_objects.  
\*   Read objects from container  
    IF line\_exists( lt\_objects[ element = '\_ATTACH\_OBJECTS' ] ).  
      DATA(ls\_att\_object) = lt\_objects[ element = '\_ATTACH\_OBJECTS' ]  .  
      DATA(lv\_value) = ls\_att\_object-value.  
      CLEAR ls\_att\_object.  
    ENDIF.  
\*   Read decision reason  
    IF NOT lv\_value IS INITIAL.  
      DATA lv\_document\_id TYPE so\_entryid.  
      lv\_document\_id = lv\_value.  
      CALL FUNCTION 'SO\_DOCUMENT\_READ\_API1'  
        EXPORTING  
          document\_id                = lv\_document\_id  
        TABLES  
          object\_content             = lt\_content  
        EXCEPTIONS  
          document\_id\_not\_exist      = 1  
          operation\_no\_authorization = 2  
          x\_error                    = 3  
          OTHERS                     = 4.  
    ENDIF.  
    IF line\_exists( lt\_content[ 1 ] ).  
      DATA(ls\_content) = lt\_content[ 1 ].  
      decision\_note = ls\_content-line.  
      CLEAR ls\_content.  
    ENDIF.  
  ENDMETHOD.  
  
  
\* <SIGNATURE>---------------------------------------------------------------------------------------+  
\* | Static Public Method ZCL\_SWF\_NOTI\_UTIL=>NOTIFY  
\* +-------------------------------------------------------------------------------------------------+  
\* | [--->] I\_SWND                         TYPE        ZST\_SWND(optional)  
\* | [--->] I\_AGENTS                       TYPE        ZTT\_AGENTS(optional)  
\* | [<---] E\_RETURN                       TYPE        BAPIRET2  
\* +--------------------------------------------------------------------------------------</SIGNATURE>  
  METHOD notify.  
  
\*   Hold agents  
    zcl\_swf\_noti\_util=>hold\_agents(  
        i\_swnd = i\_swnd  
    ).  
\*   Get user detail  
    zcl\_swf\_noti\_util=>get\_notif\_det(  
      EXPORTING  
        i\_swnd = i\_swnd  
      IMPORTING  
        e\_usrd = DATA(lt\_usrd)  
    ).  
\*    Sending email to users....  
    LOOP AT lt\_usrd ASSIGNING FIELD-SYMBOL(<fs\_usrd>).  
\*      i\_  
\*     Get email content  
      zcl\_swf\_noti\_util=>get\_email\_content(  
      EXPORTING  
        i\_swnd      = i\_swnd  
        i\_usrd      = <fs\_usrd>  
      IMPORTING  
      e\_subject     = notify\_d-subject       "Subject  
      e\_main\_text   = notify\_d-main\_text     "Text table  
            ).  
  
      notify\_d-email\_addr = CONV ad\_smtpadr( <fs\_usrd>-email ).  "Recipient (e-mail address)  
\*     BOC Abdul 08/24/2020 Adding sender email email  
      notify\_d-addr\_name  = c\_sender.  
      notify\_d-addr\_string = c\_sender\_email.  
\*     EOC Abdul 08/24/2020  
\*     Send email  
      zcl\_swf\_noti\_util=>send\_mail(  
        EXPORTING  
          i\_notify\_d = notify\_d              "Notification detail  
        IMPORTING  
          es\_return  = e\_return              "Return Parameter  
      ).  
    ENDLOOP.  
  ENDMETHOD.  
  
  
\* <SIGNATURE>---------------------------------------------------------------------------------------+  
\* | Static Public Method ZCL\_SWF\_NOTI\_UTIL=>READ\_LF\_APPROVER  
\* +-------------------------------------------------------------------------------------------------+  
\* | [--->] I\_AGENTS                       TYPE        ZTT\_AGENTS(optional)  
\* | [--->] I\_SWRD                         TYPE        ZST\_SWRD(optional)  
\* | [--->] I\_SEQ                          TYPE        ZSEQ(optional)  
\* | [--->] I\_REQID                        TYPE        TIM\_REQ\_ID(optional)  
\* | [--->] I\_REQ\_VER                      TYPE        TIM\_REQ\_VERSION(optional)  
\* | [--->] I\_INIT                         TYPE        SWP\_INITIA(optional)  
\* | [<---] E\_AGENT                        TYPE        SWP\_AGENT  
\* | [<---] E\_FINALAPP                     TYPE        ZSEQ  
\* | [<---] E\_PROCESS                      TYPE        ZPROCESS  
\* +--------------------------------------------------------------------------------------</SIGNATURE>  
  METHOD READ\_LF\_APPROVER.  
  
    DATA: LR\_APVS TYPE TABLE OF ZLR\_APVS.  
    DATA: LCL\_REQUEST TYPE REF TO if\_PT\_REQ\_REQUEST.  
    DATA: LCL\_REQUEST2 TYPE REF TO CL\_PT\_REQ\_REQUEST,  
          lv\_request type OS\_GUID.                                   "globly unique identifier  
  
\*\*&---------------------------------------------------------------------\*  
\*\*      Get instance number of workitem  
\*\*----------------------------------------------------------------------\*  
  
    """"""""""""""""""""""""""""""' GETTING APPROVER BY LEAVE TYPE""""""""""""""""""""""  
\*      COMMIT WORK.  
      WAIT UP TO 10 SECONDS.  
  
  
  
    SELECT SINGLE \*  
       FROM PTREQ\_HEADER  
       INTO @DATA(PTREQ\_HEADER)  
        WHERE REQUEST\_ID EQ @I\_REQID  
        AND VERSION\_NO EQ '1'.  
  
\*      ASSIGNING REQUEST TO GLOBLY UNIQUE IDENTIFIER  
  
      lv\_request = I\_REQID.  
  
     """"""""GETTING PERSISTENT OBJECT  
  
     LCL\_REQUEST2 ?= CA\_PT\_REQ\_HEADER=>agent->if\_os\_ca\_persistency~get\_persistent\_by\_oid( i\_oid = lv\_request ).  
  
\*    \*           Get item\_id  
    SELECT SINGLE ITEM\_INS  
      FROM PTREQ\_ITEMS  
      INTO @DATA(ITEM\_INS)  
      WHERE ITEM\_LIST\_ID = @PTREQ\_HEADER-ITEM\_LIST\_ID.  
    DATA(ITEM\_ID) = CONV TIM\_REQ\_ITEM\_ID( ITEM\_INS ).  
\*           Get request data  
    SELECT SINGLE \*  
      FROM PTREQ\_ATTABSDATA  
      INTO @DATA(PTREQ\_ATTABSDATA)  
      WHERE ITEM\_ID EQ @ITEM\_ID.  
  
"Commented by M hamza Zahid 01.04.2022  
\*    IF ( PTREQ\_ATTABSDATA-SUBTY EQ '0100' )  
\*    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0200' )  
\*    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0300' )  
\*    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0600' ).  
  
\*      I\_SWRD-PROCESS = 'LR'.  
\*      E\_PROCESS  = I\_SWRD-PROCESS.  
  
"Commented by M hamza Zahid 01.04.2022  
  
"ADDED by M hamza Zahid 01.04.2022  
    IF ( PTREQ\_ATTABSDATA-SUBTY EQ '0200' )  
    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0300' )  
    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0600' ).  
  
      I\_SWRD-PROCESS = 'LR'.  
      E\_PROCESS  = I\_SWRD-PROCESS.  
  
"ADDED by M hamza Zahid 01.04.2022  
  
"Commented by M hamza Zahid 01.04.2022  
\*    ELSEIF ( PTREQ\_ATTABSDATA-SUBTY EQ '0400' )  
\*    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0401' )  
\*    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0700' )  
\*    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0710' )  
\*    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0720' )  
\*    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0500' )  
\*    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0550' )  
\*    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0800' ).  
\*      I\_SWRD-PROCESS = 'LR2'.  
\*      E\_PROCESS  = I\_SWRD-PROCESS.  
  
  
"Commented by M hamza Zahid 01.04.2022  
  
"ADDED by M hamza Zahid 01.04.2022  
  
    ELSEIF ( PTREQ\_ATTABSDATA-SUBTY EQ '0100' )  
    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0400' )  
    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0401' )  
    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0700' )  
    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0710' )  
    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0720' )  
    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0500' )  
    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0550' )  
    OR ( PTREQ\_ATTABSDATA-SUBTY EQ '0800' ).  
  
      I\_SWRD-PROCESS = 'LR2'.  
      E\_PROCESS  = I\_SWRD-PROCESS.  
  
    ENDIF.  
  
"ADDED by M hamza Zahid 01.04.2022  
  
  
\*    SETTING INITIATOR  
  
    PTREQ\_HEADER-MOD\_USER = I\_INIT+2.  
  
    SELECT SINGLE PERNR  
    FROM PA0105 INTO @DATA(LV\_PERNR)  
    WHERE USRID EQ @PTREQ\_HEADER-MOD\_USER  
      AND ENDDA EQ @C\_ENDDA.  
  
    DATA MANAGER TYPE C.  
  
\*   Get employee position  
    SELECT SINGLE PLANS  
    FROM PA0001  
    INTO @DATA(POSITION)  
          WHERE PERNR EQ @LV\_PERNR  
          AND ENDDA EQ @C\_ENDDA.  
  
    DATA(OBJID) = POSITION.  
  
\*    Get releations from custom table  
    SELECT \*  
    FROM ZLR\_RELA  
    INTO TABLE @DATA(LR\_RELA)  
          WHERE PROCESS EQ @I\_SWRD-PROCESS  
            AND PLANS   EQ @POSITION.  
  
    IF NOT LR\_RELA IS INITIAL.  
  
      SELECT \*  
      FROM ZLR\_APVS  
      INTO TABLE LR\_APVS  
            WHERE PROCESS EQ I\_SWRD-PROCESS  
              AND PLANS   EQ POSITION.  
  
    ELSEIF ( LR\_RELA IS INITIAL ).  
  
      POSITION = '0'.  
  
      SELECT \*  
      FROM ZLR\_APVS  
      INTO TABLE LR\_APVS  
            WHERE PROCESS EQ I\_SWRD-PROCESS  
              AND PLANS   EQ POSITION.  
  
    ENDIF.  
  
    DESCRIBE TABLE LR\_APVS LINES DATA(LINES).  
    IF I\_SEQ EQ LINES.  
  
      E\_FINALAPP = ABAP\_TRUE.  
  
    ENDIF.  
\*   Get relations types  
    SELECT \*  
    FROM HRP1001  
    INTO TABLE @DATA(LT\_HRP1001)  
          WHERE OTYPE EQ @C\_OTYPE\_S  
          AND OBJID EQ @OBJID  
          AND ENDDA  EQ @C\_ENDDA.  
  
\* loop to get agents of approvers  
    READ TABLE LR\_APVS INTO DATA(LS\_LR\_APVS) WITH KEY SEQNO = I\_SEQ.  
    IF SY-SUBRC EQ 0.  
\* Read table to check relation in process  
  
  
      READ TABLE LT\_HRP1001 INTO DATA(LS\_HRP1001) WITH KEY SUBTY  = LS\_LR\_APVS-SUBTYPE .  
      IF SY-SUBRC EQ 0.  
        DATA(APPROVER\_POSITION) =  LS\_HRP1001-SOBID.  
\*     Get pernr of approver  
        SELECT SINGLE PERNR  
        FROM PA0001  
        INTO @DATA(PERNR)  
              WHERE PLANS EQ @APPROVER\_POSITION  
              AND  ENDDA EQ @C\_ENDDA.  
\*     Get user of approver  
        SELECT SINGLE USRID  
        FROM PA0105  
        INTO @DATA(USER)  
              WHERE PERNR EQ @PERNR  
              AND SUBTY EQ @C\_SUBTY\_USER  
              AND ENDDA EQ @C\_ENDDA.  
        DATA(AGENT) = CONV SWP\_AGENT( USER ).  
        CONCATENATE 'US' AGENT INTO E\_AGENT.  
  
        CALL METHOD LCL\_REQUEST2->if\_pt\_req\_request~set\_next\_processor  
          EXPORTING  
            IM\_ACTOR\_TYPE = 'P'                 " Request Participant Type  
            IM\_PLVAR      = '01'                " Plan Version  
            IM\_OTYPE      = 'P'                " Object Type  
            IM\_OBJID      = PERNR                 " Object ID  
          .  
  
        IF ( I\_SEQ GT 1 ).  
  
         CALL METHOD LCL\_REQUEST2->CLONE\_TO\_NEW.  
  
        ENDIF.  
  
  
        COMMIT WORK AND WAIT.  
      ENDIF.  
    ENDIF.  
  
  
    """""""""""""""""""""""""change status from APPROVED TO SENT""""""""""""""""""""""""""""""""  
  
    IF ( I\_SEQ GT 1 ).  
  
      CALL FUNCTION 'ENQUEUE\_EPTREQ'  
        EXPORTING  
          MODE\_PTREQ\_HEADER = 'S'  
          MANDT             = SY-MANDT  
          REQUEST\_ID        = PTREQ\_HEADER-REQUEST\_ID  
        EXCEPTIONS  
          FOREIGN\_LOCK      = 1  
          SYSTEM\_FAILURE    = 2  
          OTHERS            = 3.  
  
  
      DATA LV\_NEW\_STATUS TYPE TIM\_REQ\_STATUS.  
  
      LCL\_REQUEST2 ?= CA\_PT\_REQ\_HEADER=>agent->if\_os\_ca\_persistency~get\_persistent\_by\_oid( i\_oid = lv\_request ).  
\*  
      CALL METHOD LCL\_REQUEST2->if\_pt\_req\_request~initiate\_state\_transition  
        EXPORTING  
          im\_transfer\_event = CL\_PT\_REQ\_CONST=>c\_reqtrans\_send  
        IMPORTING  
          ex\_new\_status     = LV\_NEW\_STATUS  
        .  
\*       COMMIT WORK AND WAIT.  
  
      CALL FUNCTION 'DEQUEUE\_EPTREQ'  
        EXPORTING  
          MODE\_PTREQ\_HEADER = 'S'  
          MANDT             = SY-MANDT  
          REQUEST\_ID        = PTREQ\_HEADER-REQUEST\_ID.  
    ENDIF.  
  
  ENDMETHOD.  
  
  
\* <SIGNATURE>---------------------------------------------------------------------------------------+  
\* | Static Private Method ZCL\_SWF\_NOTI\_UTIL=>SEND\_MAIL  
\* +-------------------------------------------------------------------------------------------------+  
\* | [--->] I\_NOTIFY\_D                     TYPE        TS\_NOTIFY\_D(optional)  
\* | [<---] ES\_RETURN                      TYPE        BAPIRET2  
\* +--------------------------------------------------------------------------------------</SIGNATURE>  
  method SEND\_MAIL.  
  
\*/.. Internal table declaration  
    DATA : lt\_otf         TYPE TABLE OF itcoo,  
           lt\_pdf\_tab     TYPE TABLE OF tline,  
           lt\_content\_txt TYPE soli\_tab,  
           lt\_content\_hex TYPE solix\_tab,  
           lt\_objhead     TYPE soli\_tab.  
  
\*/.. Structure Declaration  
    DATA : ls\_otf       TYPE itcoo,  
           ls\_content   TYPE soli,  
           ls\_main\_text TYPE soli.  
  
\*/.. Class Declarations  
    DATA: go\_send\_request  TYPE REF TO  cl\_bcs,  
          go\_document      TYPE REF TO  cl\_document\_bcs,  
          lo\_recipient     TYPE REF TO  if\_recipient\_bcs,  
          lo\_bcs\_exception TYPE REF TO  cx\_bcs,  
          lo\_sender        TYPE REF TO cl\_cam\_address\_bcs.  
  
\*/.. Variables Decalrations  
    DATA : lv\_bin\_filesize TYPE i,  
           lv\_transfer\_bin TYPE sx\_boolean,  
           lv\_length       TYPE i,  
           lv\_len          TYPE so\_obj\_len,  
           lv\_sent\_to\_all  TYPE os\_boolean.  
  
\*/.. Constants Decalrations  
    CONSTANTS : c\_pdf(3)          TYPE c VALUE 'PDF',  
                c\_txt(3)          TYPE c VALUE 'TXT',  
                c\_otf(3)          TYPE c VALUE 'OTF',  
                c\_raw(3)          TYPE c VALUE 'RAW',  
                c\_error(1)        TYPE c VALUE 'E',  
                c\_success(1)      TYPE c VALUE 'S',  
                c\_success\_msg(10) TYPE c VALUE 'Email Sent',  
                c\_error\_msg(14)   TYPE c VALUE 'Email Not Sent'.  
  
\*/.. PDF Form  
    IF NOT i\_notify\_d-pdf-pdf IS INITIAL.  
  
\*/.. converting xstring into binary  
      CALL FUNCTION 'SCMS\_XSTRING\_TO\_BINARY'  
        EXPORTING  
          buffer        = i\_notify\_d-pdf-pdf "buffer  
        IMPORTING  
          output\_length = lv\_length  
        TABLES  
          binary\_tab    = lt\_content\_hex.  
  
      lv\_len = lv\_length.  
      CONDENSE lv\_len NO-GAPS.  
  
    ENDIF.  
    IF NOT i\_notify\_d-attach\_text IS INITIAL.  
      CALL FUNCTION 'SCMS\_TEXT\_TO\_BINARY'  
        IMPORTING  
          output\_length = lv\_length  
        TABLES  
          text\_tab      = i\_notify\_d-attach\_text  
          binary\_tab    = lt\_content\_hex  
        EXCEPTIONS  
          failed        = 1  
          OTHERS        = 2.  
    ENDIF.  
    TRY.  
\*/.. Create persistent send request  
        go\_send\_request = cl\_bcs=>create\_persistent( ).  
        go\_document = cl\_document\_bcs=>create\_document(  
              i\_type    = c\_raw               "RAW  
              i\_text    = i\_notify\_d-main\_text  
              i\_subject = i\_notify\_d-subject ).  
        IF NOT i\_notify\_d-attach\_text IS INITIAL.  
          IF NOT lt\_content\_hex IS INITIAL.  
            go\_document->add\_attachment(  
              i\_attachment\_type    =  c\_txt     "PDF  
              i\_attachment\_subject =  i\_notify\_d-title\_attach  
              i\_attachment\_size    =  lv\_len  
              i\_att\_content\_hex    =  lt\_content\_hex ).  
          ENDIF.  
        ENDIF.  
        IF NOT i\_notify\_d-pdf-pdf IS INITIAL.  
          IF NOT lt\_content\_hex IS INITIAL.  
            go\_document->add\_attachment(  
              i\_attachment\_type    =  c\_pdf     "PDF  
              i\_attachment\_subject =  i\_notify\_d-title\_attach  
              i\_attachment\_size    =  lv\_len  
              i\_att\_content\_hex    =  lt\_content\_hex ).  
          ENDIF.  
        ENDIF.  
\*/.. Add document object to send request  
        go\_send\_request->set\_document( go\_document ).  
        IF NOT i\_notify\_d-addr\_name IS INITIAL AND i\_notify\_d-addr\_string IS NOT INITIAL.  
          " setting sender address  
          lo\_sender    = cl\_cam\_address\_bcs=>create\_internet\_address(  
                                           EXPORTING  
                                           i\_address\_name   = i\_notify\_d-addr\_name  
                                           i\_address\_string = i\_notify\_d-addr\_string  
  
                                          ).  
          go\_send\_request->set\_sender( lo\_sender ).  
        ENDIF.  
\*/.. Add recipient (e-mail address)  
\*/.. create recipient object  
        lo\_recipient = cl\_cam\_address\_bcs=>create\_internet\_address(  
        i\_notify\_d-email\_addr ).  
  
\*/.. Add recipient object to send request  
        go\_send\_request->add\_recipient( i\_recipient = lo\_recipient ).  
  
\*        IF NOT i\_email\_addr\_cc IS INITIAL.  
\*          CLEAR lo\_recipient.  
\*  
\*\*/.. create recipient object  
\*          lo\_recipient = cl\_cam\_address\_bcs=>create\_internet\_address(  
\*          i\_email\_addr\_cc ).  
\*  
\*\*/.. Add recipient object to send request  
\*          go\_send\_request->add\_recipient( i\_recipient = lo\_recipient  
\*                                          i\_copy = 'X').  
\*  
\*        ENDIF.  
\*/.. Send document  
        lv\_sent\_to\_all = go\_send\_request->send( i\_with\_error\_screen =  
        abap\_true ).  
        COMMIT WORK.  
        IF lv\_sent\_to\_all IS INITIAL.  
          es\_return-type    = c\_error.            "E  
          es\_return-message = c\_error\_msg.        "Email not sent  
        ELSE.  
          es\_return-type    = c\_success.           "S  
          es\_return-message = c\_success\_msg.       "Email Sent  
        ENDIF.  
\*/.. Exception handling  
      CATCH cx\_bcs INTO lo\_bcs\_exception.  
    ENDTRY.  
  endmethod.  
ENDCLASS.