```
FUNCTION zscwm_mfsact_swslg_csr.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     REFERENCE(IV_LGNUM) TYPE  /SCWM/LGNUM
*"     REFERENCE(IV_PLC) TYPE  /SCWM/DE_MFSPLC
*"     REFERENCE(IV_CHANNEL) TYPE  /SCWM/DE_MFSCCH
*"     REFERENCE(IV_WTNUM) TYPE  /SCWM/TANUM
*"     REFERENCE(IV_RSRC_CNT) TYPE  /SCWM/DE_MFS_RSRC_CNT_REDU
*"     REFERENCE(IS_TELEGRAM) TYPE  /SCWM/S_MFS_TELETOTAL OPTIONAL
*"  EXPORTING
*"     REFERENCE(/SCWM/CX_MFS)
*"  RAISING
*"      /SCWM/CX_MFS
*"----------------------------------------------------------------------
*=======================================================================================*
*                              *MODIFICATION HISTORY*
*---------------------------------------------------------------------------------------*
* Mod. No.    Date      Name     Transport Number     Change Desc.                     *
*---------------------------------------------------------------------------------------*
* 1        01/12/23   Ramesh      KSDK901133        BR_MFS_15_Inclusion Auto MBRK
*                                                    logic to the cranes
*---------------------------------------------------------------------------------------*
  DATA: lt_error TYPE TABLE OF zcon_param_val.
  DATA: lt_error_code TYPE RANGE OF zcon_param_val-zvalue.

  DATA:
    gc_open    TYPE xfeld VALUE 'X',
    lt_bin     TYPE /scwm/tt_binstatus,
    ls_bin     TYPE /scwm/s_binstatus,
    lt_binobj  TYPE /scwm/tt_binstobj,
    gc_conf    TYPE xfeld VALUE 'X',
    gc_srcdata TYPE xfeld VALUE 'X',
    gc_dstdata TYPE xfeld VALUE 'X',
    gs_selcrit TYPE /scwm/s_to_selcrit_mon,
    gt_to      TYPE /scwm/tt_to_det_mon.
  DATA: lv_msgtext TYPE string,                            "#EC NEEDED
        lv_who     TYPE /scwm/de_who.
  DATA: ls_mfs_conf  TYPE /scwm/s_mfs_conf.
  DATA: lo_log       TYPE REF TO /scwm/cl_log.
  DATA:lv_vlpla TYPE /scwm/ordim_o-vlpla.
  DATA: lt_cancl   TYPE /scwm/tt_cancl,
        ls_cancel  LIKE LINE OF lt_cancl,
        ls_huid    TYPE /scwm/s_huident,
        lt_huident TYPE /scwm/tt_huident.

  DATA:lt_canc           TYPE TABLE OF ztask_log,
       ls_canc           LIKE LINE OF lt_canc,
       ls_nlpla          TYPE /scwm/ltap_nlpla, " For bin block
       ls_huident        TYPE /scwm/de_huident,
       ls_telegram       TYPE /scwm/s_mfs_teletotal,
       lt_create_hu      TYPE /scwm/tt_to_crea_hu,
       lv_severity       TYPE bapi_mtype,
       lt_bapiret        TYPE  bapirettab,
       lt_cancel_tracker TYPE TABLE OF zcanceled_task,
       ls_cancel_tracker LIKE LINE OF lt_cancel_tracker.
  DATA:lv_count TYPE i VALUE 1.
  DATA:lv_times TYPE i VALUE 4.
  DATA:lv_destbin TYPE /scwm/ordim_c-nlpla.
  DATA:lt_lagp      TYPE TABLE OF /scwm/lagp,
       ls_lagp      LIKE LINE OF lt_lagp,
       lv_empty_bin TYPE /scwm/lagp-lgpla,
       lv_new_tanum TYPE /scwm/tanum.
  DATA: lt_create    TYPE /scwm/tt_to_create_int.
*        lt_iteration TYPE TABLE OF z132_iteration,
*        ls_iteration LIKE LINE OF lt_iteration.
  DATA:lv_nlpla   TYPE /scwm/ordim_o-nlpla,
       ls_ordim_o TYPE  /scwm/ordim_o.
  DATA: ls_lbinh TYPE /scwm/e1lbinh,
        lt_lbini TYPE TABLE OF /scwm/e1lbini,
        ls_lbini TYPE /scwm/e1lbini.


  DATA:lt_ordim TYPE TABLE OF /scwm/ordim_o,
       ls_ordim LIKE LINE OF lt_ordim,
       lt_new   TYPE TABLE OF ztask_new,
       ls_new   LIKE LINE OF lt_new.
*       lv_severity TYPE bapi_mtype.

  DATA: lv_mm(2)  TYPE c,
        lv_rrr(3) TYPE c,
        lv_sss(3) TYPE c,
        lv_hh(2)  TYPE c,
        lv_dd(2)  TYPE c,
        lv_aisle  TYPE i.
  DATA: ls_der          TYPE /scwm/s_mfs_teletotal,
*             zapl_mfs_s_tele_data_swslg_der,
        lv_tanum        TYPE /scwm/tanum,
        lv_already_sent TYPE xfeld,
*        lt_canc         TYPE TABLE OF ztask_log,
*        ls_canc         LIKE LINE OF lt_canc,
        lr_ex_mfs       TYPE REF TO /scwm/cx_mfs.

  DATA:lt_132_fg TYPE TABLE OF z132_error_fg,
       ls_132_fg LIKE LINE OF lt_132_fg,
       lt_132_pm TYPE TABLE OF z132_error_pm,
       ls_132_pm LIKE  LINE OF  lt_132_pm.

  DATA: lt_ordim_o_src TYPE /scwm/tt_ordim_o,


        lt_cancl_exc   TYPE /scwm/tt_conf_exc,
        lt_ltap        TYPE /scwm/tt_ltap_vb.

  DATA:lt_crn_error TYPE TABLE OF zcrane_error_log,
       ls_crn_error LIKE LINE OF lt_crn_error.
*          *            Start of changes by vinayak
  CONSTANTS: iv_no_of_hu TYPE /scwm/lagp-anzle VALUE '2',
             lv_logpos   TYPE i VALUE '01'.
  TRY.
      CALL FUNCTION '/SCWM/MFSACT_STATUS'
        EXPORTING
          iv_lgnum    = iv_lgnum
          iv_plc      = iv_plc
          iv_channel  = iv_channel
          is_telegram = is_telegram.
    CATCH /scwm/cx_mfs.
  ENDTRY.

  CREATE OBJECT lo_log.
  IF is_telegram-assignment_id IS INITIAL.
    RETURN.
```

```
ELSE.

ENDIF.

SELECT SINGLE zvalue INTO @DATA(lv_seconds) FROM zcon_param_val WHERE lgnum EQ @iv_lgnum  AND zre
IF sy-subrc EQ 0.

ENDIF.



SELECT * INTO TABLE lt_error FROM zcon_param_val WHERE lgnum = iv_lgnum AND zreq_name = 'TASK_CAN
      AND zactive = abap_true.
IF sy-subrc EQ 0.
  lt_error_code = VALUE #( FOR ls_error IN lt_error ( sign = 'I' option = 'EQ' low = ls_error-zva
ENDIF.
IF lt_error_code[] IS NOT INITIAL.
  IF is_telegram-return_code IN lt_error_code."for other crane errors we have to cancel the tasks
    MESSAGE s003(zapl_det) WITH  is_telegram-return_code INTO lv_msgtext.
    lo_log->add_message2log( ).
    IF iv_lgnum = 'KHPM'.   " Ramesh - KSDK901133
      ls_132_pm-lgnum = iv_lgnum.
      ls_132_pm-crane_no = is_telegram-crane_number.
      lv_tanum = is_telegram-assignment_id.
      ls_132_pm-log_date = sy-datum.
      ls_132_pm-log_time = sy-uzeit.
      ls_132_pm-canceled_task = lv_tanum.
    ENDIF.
    IF iv_lgnum = 'KHFG'.
      ls_132_fg-lgnum = iv_lgnum.
      ls_132_fg-crane_no = is_telegram-crane_number.
      lv_tanum = is_telegram-assignment_id.
      ls_132_fg-log_date = sy-datum.
      ls_132_fg-log_time = sy-uzeit.
      ls_132_fg-canceled_task = lv_tanum.
    ENDIF.

    SELECT * INTO TABLE lt_ordim   FROM /scwm/ordim_o WHERE lgnum EQ iv_lgnum AND tanum EQ lv_tan

    READ TABLE lt_ordim INTO  ls_ordim WITH KEY tanum = lv_tanum.
    ls_canc-lgnum = iv_lgnum.
    ls_canc-tanum = ls_ordim-tanum.
    ls_canc-dest_bin = ls_ordim-nlpla.
    ls_canc-huident =  ls_ordim-nlenr.
    DATA(lv_hu) =  ls_ordim-nlenr.
    IF iv_lgnum = 'KHFG'.
      ls_132_fg-handling_unit = lv_hu.
    ENDIF.
    IF iv_lgnum = 'KHPM'.    " Ramesh - KSDK901133
      ls_132_pm-handling_unit = lv_hu.
    ENDIF.
    ls_canc-log_date = sy-datum.
    ls_canc-log_time = sy-uzeit.
    APPEND ls_canc TO lt_canc.
    MODIFY ztask_log FROM TABLE lt_canc.

    COMMIT WORK.


    INSERT VALUE #( sign    = 'I'
                    option  = 'EQ'
```

```
                      low     =  ls_canc-huident ) INTO TABLE gs_selcrit-r_lenr.
      CALL FUNCTION '/SCWM/TO_GET_WIP'
        EXPORTING
          iv_lgnum   = iv_lgnum
          iv_open    = gc_open
          iv_conf    = gc_conf
          iv_srcdata = gc_srcdata
          iv_dstdata = gc_dstdata
          is_selcrit = gs_selcrit
        IMPORTING
          et_to      = gt_to.

      IF sy-subrc = 0 AND gt_to IS NOT INITIAL.
        DELETE gt_to WHERE tostat NE ' '.
        DATA(lt_to) = gt_to.
        CLEAR gt_to.
        IF lt_to IS NOT INITIAL.
          READ TABLE lt_to INTO DATA(ls_to) INDEX 1.
          MOVE-CORRESPONDING ls_to TO ls_cancel.
          ls_cancel-tanum = lv_tanum.
*        ls_cancel-subst = 'X'.
          APPEND ls_cancel TO lt_cancl.
          CLEAR  ls_cancel.
        ENDIF.
      ENDIF.

**************************************************************************

      DATA:
        lv_rsrc      TYPE /scwm/de_rsrc,
        lv_move_mode TYPE /scwm/de_mfsrmm.

      DATA: ls_ordim_o_src TYPE /scwm/ordim_o,
            ls_cancl       TYPE /scwm/cancl,
            ls_rsrc        TYPE /scwm/rsrc,
            ls_cancl_exc   TYPE /scwm/s_conf_exc.
*        ls_huident     TYPE /scwm/s_huident.

      DATA:
*                  iv_lgnum       TYPE /scwm/lgnum,
*            iv_plc         TYPE /scwm/de_mfsplc,
*            is_telegram    TYPE /scwm/s_mfs_teletotal,
        io_log         TYPE REF TO /scwm/cl_log,
        iv_exccode     TYPE /scwm/de_exccode,
        it_huident     TYPE /scwm/tt_huident,
        ev_tanum_excep TYPE /scwm/tanum.
*                  RAISING /scwm/cx_mfs.

*      DATA:

*        lt_cancl       TYPE /scwm/tt_cancl,
*        lt_bapiret     TYPE bapirettab,
*        lt_cancl_exc   TYPE /scwm/tt_conf_exc,
*        lt_ltap        TYPE /scwm/tt_ltap_vb.

      FIELD-SYMBOLS: <ordim>   TYPE /scwm/ordim_o,
                     <ls_ltap> TYPE /scwm/ltap.

      CALL FUNCTION '/SCWM/TO_READ_HU'
        EXPORTING
          iv_lgnum       = iv_lgnum
          it_huident     = it_huident
        IMPORTING
          et_ordim_o_src = lt_ordim_o_src
        EXCEPTIONS
          error          = 1
          OTHERS         = 2.
      IF sy-subrc <> 0.
        /scwm/cx_mfs=>raise_exception( iv_syst = 'X' ).
      ENDIF.

      IF lt_ordim_o_src IS INITIAL.
        RETURN.
      ENDIF.

      LOOP AT lt_ordim_o_src INTO ls_ordim_o_src.
        ls_cancl-tanum = ls_ordim_o_src-tanum.
        APPEND ls_cancl TO lt_cancl.
        ev_tanum_excep = ls_ordim_o_src-tanum.
        IF iv_exccode IS NOT INITIAL.
          ls_cancl_exc-tanum     = ls_ordim_o_src-tanum.
          ls_cancl_exc-exccode   = iv_exccode.
          ls_cancl_exc-buscon    = wmegc_buscon_mf3.
          ls_cancl_exc-exec_step = wmegc_execstep_a0.
          APPEND ls_cancl_exc TO lt_cancl_exc.
        ENDIF.
      ENDLOOP.

**************************************************************************


      CALL FUNCTION '/SCWM/TO_CANCEL'
        EXPORTING
          iv_lgnum    = iv_lgnum
          it_cancl    = lt_cancl
        IMPORTING
          ev_severity = lv_severity.
      COMMIT WORK.



      WAIT UP TO lv_seconds SECONDS.
      IF lv_severity EQ 'I'.
        READ TABLE lt_cancl INTO ls_cancel INDEX 1.
        MESSAGE s001(zapl_det) WITH  ls_cancel-tanum INTO lv_msgtext.
        lo_log->add_message2log( ).
        IF iv_lgnum = 'KHPM'.      " Ramesh - KSDK901133
          ls_132_pm-description = 'TASK IS CANCELLED'.
          APPEND ls_132_pm TO lt_132_pm.
          MODIFY z132_error_pm FROM TABLE lt_132_pm.
          COMMIT WORK.
        ENDIF.
        IF iv_lgnum = 'KHFG'.
          ls_132_fg-description = 'TASK IS CANCELLED'.
          APPEND ls_132_fg TO lt_132_fg.
          MODIFY z132_error_fg FROM TABLE lt_132_fg.
          COMMIT WORK.
        ENDIF.

      ENDIF.

      UPDATE zcon_param_val SET zactive = abap_true  WHERE lgnum = iv_lgnum AND zvalue = is_telegra
```

```
    COMMIT WORK.

  ENDIF." for other errors of crane end
ENDIF.
IF is_telegram-return_code EQ '138'.
  DO lv_times TIMES.
    SELECT SINGLE vlpla INTO lv_vlpla FROM /scwm/ordim_o WHERE lgnum = iv_lgnum AND tanum = is_te
    IF sy-subrc = 0.
      lv_times = 0.
      UPDATE /scwm/lagp
              SET skzua = abap_true skzue = abap_true
                WHERE lgnum = iv_lgnum
                AND  lgpla = lv_vlpla.
      SELECT SINGLE  guid_loc INTO @DATA(ls_loc) FROM /scwm/lagp WHERE lgnum = @iv_lgnum AND lgpl
      IF sy-subrc EQ 0.

        ls_bin-objnr = ls_loc.
        ls_bin-stat = 'E0001'.
        ls_bin-inact = abap_true.
        ls_bin-chgnr = 000.
        ls_bin-mod = 'U'.
        ls_bin-incat_old = abap_false.
        APPEND ls_bin TO lt_bin.

        ls_bin-objnr = ls_loc.
        ls_bin-stat = 'E0007'.
        ls_bin-inact = abap_false.
        ls_bin-chgnr = 000.
        ls_bin-mod = 'I'.
        ls_bin-incat_old = abap_true.
        APPEND ls_bin TO lt_bin.


      ENDIF.

      CALL FUNCTION '/SCWM/BIN_STATUS_UPDATE'
        EXPORTING
          it_binstatus = lt_bin
          it_binstobj  = lt_binobj.

    ELSEIF sy-subrc <> 0.
      SELECT SINGLE vlpla  INTO lv_vlpla FROM /scwm/ordim_c WHERE lgnum = iv_lgnum AND tanum = is
      IF sy-subrc = 0.
        lv_times = 0.
        UPDATE /scwm/lagp
          SET skzua = abap_true  skzue = abap_true
          WHERE lgnum = iv_lgnum
            AND  lgpla = lv_vlpla.
        SELECT SINGLE  guid_loc INTO ls_loc FROM /scwm/lagp WHERE lgnum = iv_lgnum AND lgpla = lv
        IF sy-subrc EQ 0.

          ls_bin-objnr = ls_loc.
          ls_bin-stat = 'E0001'.
          ls_bin-inact = abap_true.
          ls_bin-chgnr = 000.
          ls_bin-mod = 'U'.
          ls_bin-incat_old = abap_false.
          APPEND ls_bin TO lt_bin.

          ls_bin-objnr = ls_loc.
          ls_bin-stat = 'E0007'.
          ls_bin-inact = abap_false.
          ls_bin-chgnr = 000.
          ls_bin-mod = 'I'.
          ls_bin-incat_old = abap_true.
          APPEND ls_bin TO lt_bin.

        ENDIF.

        CALL FUNCTION '/SCWM/BIN_STATUS_UPDATE'
          EXPORTING
            it_binstatus = lt_bin
            it_binstobj  = lt_binobj.

      ENDIF.

    ENDIF.
  ENDDO.
ENDIF.


IF is_telegram-return_code EQ '132'.
  MESSAGE s003(zapl_det) WITH  is_telegram-return_code INTO lv_msgtext.
  lo_log->add_message2log( ).
  IF iv_lgnum = 'KHPM'.      " Ramesh - KSDK901133
    ls_132_pm-lgnum = iv_lgnum.
    ls_132_pm-crane_no = is_telegram-crane_number.
    lv_tanum = is_telegram-assignment_id.
    ls_132_pm-log_date = sy-datum.
    ls_132_pm-log_time = sy-uzeit.
    ls_132_pm-canceled_task = lv_tanum.
  ENDIF.
  IF iv_lgnum = 'KHFG'.
    ls_132_fg-lgnum = iv_lgnum.
    ls_132_fg-crane_no = is_telegram-crane_number.
    lv_tanum = is_telegram-assignment_id.
    ls_132_fg-log_date = sy-datum.
    ls_132_fg-log_time = sy-uzeit.
    ls_132_fg-canceled_task = lv_tanum.
  ENDIF.

  SELECT * INTO TABLE lt_ordim   FROM /scwm/ordim_o WHERE lgnum EQ iv_lgnum AND tanum EQ lv_tanum

  READ TABLE lt_ordim INTO  ls_ordim WITH KEY tanum = lv_tanum.
  ls_canc-lgnum = iv_lgnum.
  ls_canc-tanum = ls_ordim-tanum.
  ls_canc-dest_bin = ls_ordim-nlpla.
  ls_canc-huident =  ls_ordim-nlenr.
  lv_hu =  ls_ordim-nlenr.
  IF iv_lgnum = 'KHFG'.
    ls_132_fg-handling_unit = lv_hu.
  ENDIF.
  IF iv_lgnum = 'KHPM'.      " Ramesh - KSDK901133
    ls_132_pm-handling_unit = lv_hu.
  ENDIF.
  ls_canc-log_date = sy-datum.
  ls_canc-log_time = sy-uzeit.
  APPEND ls_canc TO lt_canc.
  MODIFY ztask_log FROM TABLE lt_canc.

  COMMIT WORK.
```

```
    INSERT VALUE #( sign    = 'I'
                    option  = 'EQ'
                    low     =  ls_canc-huident ) INTO TABLE gs_selcrit-r_lenr.
    CALL FUNCTION '/SCWM/TO_GET_WIP'
      EXPORTING
        iv_lgnum   = iv_lgnum
        iv_open    = gc_open
        iv_conf    = gc_conf
        iv_srcdata = gc_srcdata
        iv_dstdata = gc_dstdata
        is_selcrit = gs_selcrit
      IMPORTING
        et_to      = gt_to.

    IF sy-subrc = 0 AND gt_to IS NOT INITIAL.
      DELETE gt_to WHERE tostat NE ' '.
      lt_to = gt_to.
      CLEAR gt_to.
      IF lt_to IS NOT INITIAL.
        READ TABLE lt_to INTO ls_to INDEX 1.
        MOVE-CORRESPONDING ls_to TO ls_cancel.
        ls_cancel-tanum = lv_tanum.
*        ls_cancel-subst = 'X'.
        APPEND ls_cancel TO lt_cancl.
        CLEAR  ls_cancel.
      ENDIF.
    ENDIF.

    MESSAGE s007(zapl_det) WITH  ls_cancel-tanum INTO lv_msgtext.
    lo_log->add_message2log( ).
    UPDATE zcon_param_val SET zactive = abap_true  WHERE lgnum = iv_lgnum AND zvalue = is_telegram-
    COMMIT WORK.

    CALL FUNCTION '/SCWM/TO_CANCEL'
      EXPORTING
        iv_lgnum    = iv_lgnum
        it_cancl    = lt_cancl
      IMPORTING
        ev_severity = lv_severity.
    COMMIT WORK.
    WAIT UP TO lv_seconds SECONDS.
    IF lv_severity EQ 'I'.
      READ TABLE lt_cancl INTO ls_cancel INDEX 1.
      MESSAGE s001(zapl_det) WITH  ls_cancel-tanum INTO lv_msgtext.
      lo_log->add_message2log( ).
      IF iv_lgnum = 'KHPM'.      " Ramesh - KSDK901133
        ls_132_pm-description = 'TASK IS CANCELLED'.
        APPEND ls_132_pm TO lt_132_pm.
        MODIFY z132_error_pm FROM TABLE lt_132_pm.
        COMMIT WORK.
      ENDIF.
      IF iv_lgnum = 'KHFG'.
        ls_132_fg-description = 'TASK IS CANCELLED'.
        APPEND ls_132_fg TO lt_132_fg.
        MODIFY z132_error_fg FROM TABLE lt_132_fg.
        COMMIT WORK.
      ENDIF.

    ENDIF.
```

```
*    ***        Bin Block
    UPDATE /scwm/lagp
      SET skzua = abap_true skzue = abap_true
    WHERE lgnum = iv_lgnum
    AND  lgpla = ls_ordim-nlpla.


    SELECT SINGLE  guid_loc INTO ls_loc FROM /scwm/lagp WHERE lgnum = iv_lgnum AND lgpla = ls_ordim
    IF sy-subrc EQ 0.

      ls_bin-objnr = ls_loc.
      ls_bin-stat = 'E0001'.
      ls_bin-inact = 'X'.
      ls_bin-chgnr = 000.
      ls_bin-mod = 'U'.
      ls_bin-inact = abap_true.
      APPEND ls_bin TO lt_bin.

      ls_bin-objnr = ls_loc.
      ls_bin-stat = 'E0006'.
      ls_bin-inact = abap_false.
      ls_bin-chgnr = 000.
      ls_bin-mod = 'I'.
      ls_bin-incat_old = abap_true.
      APPEND ls_bin TO lt_bin.

      ls_bin-objnr = ls_loc.
      ls_bin-stat = 'IBI01'.
      ls_bin-inact = abap_false.
      ls_bin-chgnr = 000.
      ls_bin-mod = 'U'.
      ls_bin-incat_old = abap_true.
      APPEND ls_bin TO lt_bin.

      ls_bin-objnr = ls_loc.
      ls_bin-stat = 'IBI02'.
      ls_bin-inact = abap_false.
      ls_bin-chgnr = 000.
      ls_bin-mod = 'U'.
      ls_bin-incat_old = abap_true.
      APPEND ls_bin TO lt_bin.

    ENDIF.

    CALL FUNCTION '/SCWM/BIN_STATUS_UPDATE'
      EXPORTING
        it_binstatus = lt_bin
        it_binstobj  = lt_binobj.


****Bin Block,
    CLEAR: ls_132_pm,ls_132_fg,lv_msgtext,lt_132_pm,lt_132_fg.
    MESSAGE s002(zapl_det) WITH  ls_ordim-nlpla INTO lv_msgtext.
    lo_log->add_message2log( ).
    IF iv_lgnum = 'KHFG'.
      CONCATENATE ls_ordim-nlpla 'bin is blocked FG' INTO lv_msgtext.
      ls_132_fg-lgnum = iv_lgnum.
      ls_132_fg-crane_no = is_telegram-crane_number.
      ls_132_fg-handling_unit = is_telegram-huident.
      ls_132_fg-log_date = sy-datum.
      ls_132_fg-log_time = sy-uzeit.
      ls_132_fg-description = lv_msgtext.
      APPEND ls_132_fg TO lt_132_fg.
      MODIFY z132_error_fg FROM TABLE lt_132_fg.
      COMMIT WORK.
    ENDIF.
    IF iv_lgnum = 'KHPM'.    " Ramesh - KSDK901133
      CONCATENATE ls_ordim-nlpla 'bin is blocked PM' INTO lv_msgtext.
      ls_132_pm-lgnum = iv_lgnum.
      ls_132_pm-crane_no = is_telegram-crane_number.
      ls_132_pm-handling_unit = is_telegram-huident.
      ls_132_pm-log_date = sy-datum.
      ls_132_pm-log_time = sy-uzeit.
      ls_132_pm-description = lv_msgtext.
      APPEND ls_132_pm TO lt_132_pm.
      MODIFY z132_error_pm FROM TABLE lt_132_pm.
      COMMIT WORK.
    ENDIF.

  ENDIF.

  IF is_telegram-return_code EQ '137' AND is_telegram-assignment_id IS NOT INITIAL.

    lv_tanum = is_telegram-assignment_id.
    SELECT SINGLE * FROM /scwm/ordim_o INTO @DATA(ls_ordim_137) WHERE lgnum EQ @iv_lgnum AND tanum
    IF sy-subrc = 0.
      ls_huid-lgnum = iv_lgnum.
      ls_huid-vhi   = wmegc_vhi_real.
      ls_huid-huident = ls_ordim_137-nlenr.
      APPEND ls_huid TO lt_huident.

      CALL FUNCTION '/SCWM/TO_READ_HU'
        EXPORTING
          iv_lgnum       = iv_lgnum
          it_huident     = lt_huident
        IMPORTING
          et_ordim_o_src = lt_ordim_o_src
        EXCEPTIONS
          error          = 1
          OTHERS         = 2.
      IF sy-subrc <> 0.
        /scwm/cx_mfs=>raise_exception( iv_syst = 'X' ).
      ENDIF.

    ENDIF.

    IF lt_ordim_o_src IS INITIAL.
      RETURN.
    ENDIF.

    LOOP AT lt_ordim_o_src INTO ls_ordim_o_src.
      ls_cancl-tanum = ls_ordim_o_src-tanum.
      APPEND ls_cancl TO lt_cancl.
    ENDLOOP.

    CALL FUNCTION '/SCWM/TO_CANCEL'
      EXPORTING
        iv_lgnum       = iv_lgnum
*       iv_subst       = 'X'
        iv_update_task = ' '
        iv_commit_work = ' '
        it_cancl       = lt_cancl
        it_cancl_exc   = lt_cancl_exc
      IMPORTING
        et_bapiret     = lt_bapiret
        ev_severity    = lv_severity.

    COMMIT WORK AND WAIT.
    /scwm/cl_tm=>cleanup( ).

    CLEAR: ls_ordim_o.

    READ TABLE lt_ordim_o_src INTO ls_ordim_o WITH KEY flghuto = 'X'.
    IF sy-subrc = 0.
      CLEAR: ls_lbinh, ls_lbini, lt_lbini.
      ls_lbinh-lgnum = iv_lgnum.
      ls_lbinh-lgtyp = ls_ordim_o-vltyp.
      ls_lbinh-block = abap_true.
      ls_lbini-lgpla = ls_ordim_o-vlpla.
      ls_lbini-skzua = abap_true.
      ls_lbini-skzue = abap_true.
      APPEND ls_lbini TO lt_lbini.

      CALL FUNCTION '/SCWM/BLOCK_BINS'
        EXPORTING
          lbinh             = ls_lbinh
        TABLES
          lbini             = lt_lbini
        EXCEPTIONS
          consistency_check = 1
          OTHERS            = 2.
      IF sy-subrc = 0.
        COMMIT WORK AND WAIT.
      ENDIF.

      lv_vlpla = ls_ordim_o-vlpla.

      SELECT SINGLE  guid_loc INTO @DATA(lv_loc) FROM /scwm/lagp WHERE lgnum = @iv_lgnum AND lgpla
      IF sy-subrc EQ 0.
        CLEAR:ls_bin.
        ls_bin-objnr = lv_loc.
        ls_bin-stat = 'E0001'.
        ls_bin-inact = abap_true.
        ls_bin-chgnr = 000.
        ls_bin-mod = 'U'.
        ls_bin-incat_old = abap_false.
        APPEND ls_bin TO lt_bin.

        CLEAR:ls_bin.
        ls_bin-objnr = lv_loc.
        ls_bin-stat = 'E0002'.
        ls_bin-inact = abap_true.
        ls_bin-chgnr = 000.
        ls_bin-mod = 'U'.
        ls_bin-incat_old = abap_false.
        APPEND ls_bin TO lt_bin.

        CLEAR:ls_bin.
        ls_bin-objnr = lv_loc.
        ls_bin-stat = 'E0003'.
        ls_bin-inact = abap_true.
        ls_bin-chgnr = 000.
        ls_bin-mod = 'U'.
```

```
        ls_bin-incat_old = abap_false.
        APPEND ls_bin TO lt_bin.

        CLEAR:ls_bin.
        ls_bin-objnr = lv_loc.
        ls_bin-stat = 'E0005'.
        ls_bin-inact = abap_true.
        ls_bin-chgnr = 000.
        ls_bin-mod = 'U'.
        ls_bin-incat_old = abap_false.
        APPEND ls_bin TO lt_bin.

        CLEAR:ls_bin.
        ls_bin-objnr = lv_loc.
        ls_bin-stat = 'E0006'.
        ls_bin-inact = abap_true.
        ls_bin-chgnr = 000.
        ls_bin-mod = 'U'.
        ls_bin-incat_old = abap_false.
        APPEND ls_bin TO lt_bin.

        CLEAR:ls_bin.
        ls_bin-objnr = lv_loc.
        ls_bin-stat = 'E0007'.
        ls_bin-inact = abap_true.
        ls_bin-chgnr = 000.
        ls_bin-mod = 'U'.
        ls_bin-incat_old = abap_false.
        APPEND ls_bin TO lt_bin.


        CLEAR:ls_bin.
        ls_bin-objnr = lv_loc.
        ls_bin-stat = 'E0004'.
        ls_bin-inact = abap_false.
        ls_bin-chgnr = 000.
        ls_bin-mod = 'I'.
        ls_bin-incat_old = abap_true.
        APPEND ls_bin TO lt_bin.
      ENDIF.
      CALL FUNCTION '/SCWM/BIN_STATUS_UPDATE'
        EXPORTING
          it_binstatus = lt_bin
          it_binstobj  = lt_binobj.
      COMMIT WORK.
    ENDIF.

    ls_telegram = is_telegram.
    ls_telegram-dest = ls_ordim_o-nlpla.

* Determine resource
    IF is_telegram-cp IS NOT INITIAL OR
       is_telegram-dest IS NOT INITIAL.
*   Even if resource is not filled in telegram we have to
*     check if there is an resource involved.
*     But this is only possible with an open WT to get the queue.
      READ TABLE lt_ordim_o_src ASSIGNING <ordim> WITH KEY flghuto = 'X'. "#EC CI_NOORDER
      IF <ordim>-queue IS NOT INITIAL.
        CALL FUNCTION '/SCWM/MFS_GET_RSRC_BY_QUEUE'
          EXPORTING
            iv_lgnum = iv_lgnum
            iv_queue = <ordim>-queue
          IMPORTING
            es_rsrc  = ls_rsrc.
        IF ls_rsrc-rsrc IS NOT INITIAL.
          lv_rsrc = ls_rsrc-rsrc.
        ENDIF.
      ENDIF.
    ELSEIF is_telegram-rsrc IS NOT INITIAL.
      lv_rsrc = is_telegram-rsrc.
    ENDIF.

    LOOP AT lt_ordim_o_src INTO ls_ordim_o_src
      WHERE flghuto IS INITIAL.

      IF is_telegram-rsrc IS NOT INITIAL.
        lv_rsrc = is_telegram-rsrc.
      ELSEIF ls_ordim_o_src-prsrc IS NOT INITIAL.
        lv_rsrc = ls_ordim_o_src-prsrc.
      ELSEIF ls_ordim_o_src-srsrc IS NOT INITIAL.
        lv_rsrc = ls_ordim_o_src-srsrc.
      ELSEIF ls_ordim_o_src-drsrc IS NOT INITIAL.
        lv_rsrc = ls_ordim_o_src-drsrc.
      ENDIF.

      CALL FUNCTION '/SCWM/TO_CREATE_BIN_DENIAL'
        EXPORTING
          iv_lgnum       = iv_lgnum
          iv_tanum       = ls_ordim_o_src-tanum
          iv_anfme       = ls_ordim_o_src-vsola
          iv_altme       = ls_ordim_o_src-altme
          iv_prsrc       = lv_rsrc
          iv_update_task = ' '
          iv_commit_work = 'X'
        IMPORTING
          et_ltap_vb     = lt_ltap.

      LOOP AT lt_ltap ASSIGNING <ls_ltap>.
        MESSAGE i199(/scwm/mfs) WITH <ls_ltap>-tanum INTO gv_msgtext.
        io_log->add_message( ip_row = 0 ).
      ENDLOOP.

    ENDLOOP.

  ENDIF.


ENDFUNCTION.
```