

*&-----
---*

* Name :Outbound Bank IDOC Trigger
* Created By :
* Created On :13/03/2019
* Purpose :To trigger bank keys delta replication to EC and ECP
* Transport :
* Code Review : <USERID> <DATE mm/dd/yyyy>

*&-----
---*

* Revisions History:

*&-----
---*

* Number	Date	Author
* _____	_____	_____

*&-----
---*

*&-----
---*

*& Report ZHCM_BANKKEY_SELECT

*&-----
---*

*&
*&

*&-----
---*

REPORT zhcmsfsfp_bankkeysreplication MESSAGE-ID z_hcm_msgclass.

TYPES: BEGIN OF gty_cdhdr,
 objectid TYPE cdhdr-objectid,
 changenr TYPE cdhdr-changenr,
 username TYPE cdhdr-username,
 update TYPE cdhdr-update,
 objectclass TYPE cdhdr-objectclas,
END OF gty_cdhdr.

TYPES: BEGIN OF gty_bank,
 banks TYPE banks,
 bankl TYPE bankl,
 brnch TYPE brnch,
END OF gty_bank.

TABLES : bnka.

DATA: gt_cdhdr TYPE TABLE OF gty_cdhdr,
 gs_cdhdr TYPE gty_cdhdr,
 gs_bank TYPE gty_bank,
 gt_bank TYPE STANDARD TABLE OF gty_bank,
 gr_bank_banks TYPE RANGE OF banks,
 gs_bank_banks LIKE LINE OF gr_bank_banks,

```

gr_bank_bankl TYPE RANGE OF bankl,
gs_bank_bankl LIKE LINE OF gr_bank_bankl,
gd_logsys     TYPE tbdls-logsys,
gv_date       TYPE sy-datum,
gv_time       TYPE sy-uzeit,
gs_tvarvc     TYPE tvarvc,
lv_temp       TYPE sy-subrc.

```

```

CONSTANTS: gc_bank TYPE char4 VALUE 'BANK',
            gc_x    TYPE c   VALUE 'X'.

```

```

SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE text-001.

```

```

PARAMETERS: p_logsys TYPE tbdls-logsys OBLIGATORY DEFAULT
'XLUCLNT100',

```

```

        p_date  TYPE sy-datum MODIF ID dat,
        p_time  TYPE sy-uzeit MODIF ID tim,
        rb_r1   RADIOBUTTON GROUP rad1 USER-COMMAND rad1,
        rb_r2   RADIOBUTTON GROUP rad1 DEFAULT 'X',
        rb_r3   RADIOBUTTON GROUP rad1.

```

```

SELECT-OPTIONS s_bkey FOR bnka-bankl.

```

```

PARAMETERS ch_emp TYPE abap_bool AS CHECKBOX DEFAULT abap_true.
SELECTION-SCREEN END OF BLOCK b1.

```

```

AT SELECTION-SCREEN.

```

```

    IF rb_r1 = gc_x.
        gv_date = p_date.
        gv_time = p_time.
    ELSEIF rb_r2 = gc_x.
        SELECT SINGLE * FROM tvarvc INTO gs_tvarvc
            WHERE name = 'ZHR_BANK_LASTSUCCRUNDATE'.
        IF sy-subrc <> 0.
            MESSAGE e159.
        ELSE.
            gv_date = gs_tvarvc-low.
        ENDIF.
        CLEAR gs_tvarvc.

```

```

        SELECT SINGLE * FROM tvarvc INTO gs_tvarvc
            WHERE name = 'ZHR_BANK_LASTSUCCRUNTIME'.

```

```

        IF sy-subrc <> 0.
            MESSAGE e160.
        ELSE.
            gv_time = gs_tvarvc-low.
        ENDIF.
        CLEAR gs_tvarvc.

```

```

    ENDIF.

```

```

AT SELECTION-SCREEN OUTPUT.

```

```

    LOOP AT SCREEN.

```

```

        IF rb_r2 = gc_x.
            IF screen-name = 'P_DATE'.
                screen-input = '0'.
            ENDIF.
        ENDIF.
    ENDLOOP.

```

```

ELSEIF screen-name = 'P_TIME'.
    screen-input = '0'.
ELSEIF screen-name CS 'S_BKEY'.
    screen-active = '0'.
ENDIF.
ELSEIF rb_r1 = gc_x.
    IF screen-name CS 'S_BKEY'.
        screen-active = '0'.
    ENDIF.
ELSEIF rb_r3 = gc_x.
    IF screen-name = 'P_DATE'.
        screen-input = '0'.
    ELSEIF screen-name = 'P_TIME'.
        screen-input = '0'.
    ELSEIF screen-name CS 'S_BKEY'.
        screen-active = '1'.
    ENDIF.
ENDIF.
MODIFY SCREEN.
ENDLOOP.

START-OF-SELECTION.
    gd_logsys = p_logsys.

    IF rb_r3 <> gc_x.
        SELECT objectid                                "This field contain all the
        changes done in FI01,FI02 and FI06
            changenr
            username
            udate
        FROM cdhdr
        INTO TABLE gt_cdhdr
        WHERE objectclas = gc_bank
            AND ( ( udate GT gv_date )
                OR ( udate EQ gv_date AND utime GT gv_time ) ). " It will
take the value after the date and time specified.
        IF sy-subrc NE 0.
            IF sy-batch <> abap_true.
                MESSAGE e157.
            ELSE.
                WRITE: / 'No changes to the existing bank table for the
selected date'.
                LEAVE LIST-PROCESSING.
            ENDIF.
        ELSE.
            SORT gt_cdhdr BY udate.
            LOOP AT gt_cdhdr INTO gs_cdhdr.
                gs_bank-banks = gs_cdhdr-objectid+3(2). "Here we get Bank
country
                gs_bank-bankl = gs_cdhdr-objectid+6.      "Here we get Bank
key
                gs_bank_banks-sign = 'I'.
                gs_bank_banks-option = 'EQ'.
                gs_bank_banks-low = gs_bank-banks.

```

```

APPEND gs_bank_banks TO gr_bank_banks.
SORT gr_bank_banks.
DELETE ADJACENT DUPLICATES FROM gr_bank_banks.

gs_bank_bankl-sign = 'I'.
gs_bank_bankl-option = 'EQ'.
gs_bank_bankl-low = gs_bank-bankl.
APPEND gs_bank_bankl TO gr_bank_bankl.
SORT gr_bank_bankl.
DELETE ADJACENT DUPLICATES FROM gr_bank_bankl.
ENDLOOP.
ENDIF.
ELSEIF rb_r3 EQ gc_x.
LOOP AT s_bkey.
gs_bank_bankl-sign = 'I'.
gs_bank_bankl-option = 'EQ'.
gs_bank_bankl-low = s_bkey-low.
APPEND gs_bank_bankl TO gr_bank_bankl.
CLEAR gs_bank_bankl.
ENDLOOP.
ENDIF.

* To exclude non Employee banks
IF gr_bank_bankl IS NOT INITIAL.
IF ch_emp EQ abap_true.
SELECT banks
      bankl
      brnch
FROM bnka
INTO TABLE gt_bank
WHERE banks IN gr_bank_banks
AND bankl IN gr_bank_bankl.
IF sy-subrc = 0.
LOOP AT gt_bank INTO gs_bank WHERE brnch <> 'Employee Only'.
DELETE gr_bank_banks WHERE low EQ gs_bank-banks.
DELETE gr_bank_bankl WHERE low EQ gs_bank-bankl.
ENDLOOP.
ENDIF.
ENDIF.
ENDIF.

CLEAR sy-subrc.
IF ( gr_bank_banks IS NOT INITIAL ) OR ( gr_bank_bankl IS NOT
INITIAL ).
SUBMIT rfbank_ale " This Submit
statement calls T-Code 'FI08' and provides the value.
      WITH banks IN gr_bank_banks
      WITH bankl IN gr_bank_bankl
      WITH t_rec EQ gd_logsys
      WITH testrun EQ ''
      AND RETURN.
lv_temp = sy-subrc.
ENDIF.

```

```
IF lv_temp EQ 0 AND rb_r3 <> gc_x..
  MESSAGE i158.      "Successfully submitted.
  gs_tvarvc-mandt = sy-mandt.
  gs_tvarvc-name = 'ZHR_BANK_LASTSUCCRUNDATE'.
  gs_tvarvc-numb = '0000'.
  gs_tvarvc-type = 'S'.
  gs_tvarvc-sign = 'I'.
  gs_tvarvc-opti = 'EQ'.
  gs_tvarvc-low = sy-datum.
  UPDATE tvarvc FROM gs_tvarvc.

  gs_tvarvc-low = sy-uzeit.
  gs_tvarvc-name = 'ZHR_BANK_LASTSUCCRUNTIME'.
  UPDATE tvarvc FROM gs_tvarvc.
ENDIF.

CALL FUNCTION 'DB_COMMIT'.
CALL FUNCTION 'DEQUEUE_ALL'.
COMMIT WORK.
```