

METHOD if\_cod\_sls\_se\_simulate\_order~adjust\_import\_data.

\*-----\*

\* Method : IF\_COD\_SLS\_SE\_SIMULATE\_ORDER~ADJUST\_IMPORT\_DATA

\* Description : Adding the logic to replace the customer number in incoming data if customer not

\* found in S4 because Prospects only exist in C4C not in S4.

\* Dev Scope : Pricing determination doesn't work for prospect since prospects are only exist only

\* in C4C and not in S4. To achieve the requirement replacing the Customer number in

\* the incoming data with default one if the customer number is not found in S4.

\*-----\*

CONSTANTS :

lc\_intg\_user TYPE zde\_key VALUE 'C4C\_INTG\_USER',

lc\_def\_partner TYPE zde\_key VALUE 'DEFAULT\_OPPT\_QUOTE\_PARTNER',

lc\_cust\_role TYPE but100-rltyp VALUE 'FLCU01',

lc\_mesg\_cls TYPE arbgb VALUE 'ZM\_INTG\_MESSAGES'.

TYPES : BEGIN OF lty\_hardcd,

hardcode\_key TYPE zct\_avoid\_hardcd-hardcode\_key,

hardcode\_value TYPE zct\_avoid\_hardcd-hardcode\_value,

END OF lty\_hardcd.

TYPES : BEGIN OF lty\_but100,

partner TYPE but100-partner,

rltyp TYPE but100-rltyp,

END OF lty\_but100.

DATA :

lro\_key TYPE RANGE OF zct\_avoid\_hardcd-hardcode\_key,

```
lrs_key LIKE LINE OF lro_key,  
lt_hardcd TYPE TABLE OF lty_hardcd,  
lt_but100 TYPE TABLE OF lty_but100,  
ls_hardcd TYPE lty_hardcd,  
ls_bapiret2 TYPE bapiret2,  
lv_bp_flag TYPE boolean,  
lv_partner TYPE but000-partner.
```

FIELD-SYMBOLS: <lfs\_partners> TYPE bapiparnr.

\* Clear the ITAB's.

```
REFRESH : lt_hardcd[],lt_but100[].
```

\* Check the Partners ITAB is initial or not.

```
CHECK ct_order_partners[] IS NOT INITIAL.
```

\* Alpha input conversion for customer number

```
LOOP AT ct_order_partners ASSIGNING <lfs_partners> .
```

```
IF <lfs_partners>-partn_num IS NOT INITIAL.
```

```
CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
```

```
EXPORTING
```

```
input = <lfs_partners>-partn_num
```

```
IMPORTING
```

```
output = <lfs_partners>-partn_num.
```

```
ENDIF.
```

```
ENDLOOP.
```

\* Prepare the Range table.

```
CLEAR : Irs_key ,Iro_key.
```

```
Irs_key-sign = 'I'.
```

```
Irs_key-option = 'EQ'.
```

```
Irs_key-low = Ic_intg_user .
```

```
APPEND Irs_key TO Iro_key.
```

```
CLEAR : Irs_key.
```

```
Irs_key-sign = 'I'.
```

```
Irs_key-option = 'EQ'.
```

```
Irs_key-low = Ic_def_partner.
```

```
APPEND Irs_key TO Iro_key.
```

```
IF Iro_key[] IS NOT INITIAL.
```

\* Read the entries from Hard code table.

```
SELECT hardcode_key
```

```
    hardcode_value FROM zct_avoid_hardcd
```

```
    INTO TABLE It_hardcd
```

```
    WHERE hardcode_key IN Iro_key.
```

```
IF sy-subrc EQ 0.
```

```
    SORT It_hardcd BY hardcode_key ASCENDING.
```

```
ENDIF.
```

```
ENDIF.
```

```
CLEAR : Is_hardcd.
```

\* Check if the User is Integration User or not.

```
READ TABLE It_hardcd INTO Is_hardcd WITH KEY hardcode_key = Ic_intg_user BINARY SEARCH.
```

```
IF sy-subrc NE 0 OR Is_hardcd-hardcode_value NE sy-uname ##USER_OK.
```

\* If User is not a Integration user, exit from method.

RETURN.

ENDIF.

\* Read the partner roles.

SELECT partner

    rltyp FROM but100

        INTO TABLE It\_but100

        FOR ALL ENTRIES IN ct\_order\_partners

        WHERE partner = ct\_order\_partners-partn\_numb

        AND rltyp = lc\_cust\_role. "Customer

IF sy-subrc EQ 0.

\* Sort the ITAB.

    SORT It\_but100 BY partner ASCENDING.

ENDIF.

CLEAR :lv\_bp\_flag, ls\_bapiret2, lv\_partner , ls\_hardcd.

\* Read the Default partner from Hardcode table.

READ TABLE It\_hardcd INTO ls\_hardcd WITH KEY hardcode\_key = lc\_def\_partner BINARY SEARCH.

IF sy-subrc EQ 0.

    lv\_partner = ls\_hardcd-hardcode\_value.

    UNPACK lv\_partner TO lv\_partner.

ELSE.

    lv\_bp\_flag = abap\_true.

ENDIF.

\* Loop on all Partners.

LOOP AT ct\_order\_partners ASSIGNING <lfs\_partners> .

\* Check partner have Customer role or not. If not replace with default partner.

```
READ TABLE lt_but100 WITH KEY partner = <lfs_partners>-partn_numm BINARY SEARCH  
TRANSPORTING NO FIELDS.
```

```
IF sy-subrc NE 0.
```

```
  <lfs_partners>-partn_numm = lv_partner.
```

```
ELSE.
```

```
  lv_bp_flag = abap_false.
```

```
ENDIF.
```

```
ENDLOOP.
```

\* Throw exception and return message in the case if partner not exist in the system and Dummy customer is not maintained

```
IF lv_bp_flag = abap_true.
```

\* Add the message, if Dummy customer not maintained in table "ZCT\_AVOID\_HARDCD".

```
CLEAR :ls_bapiret2.
```

```
ls_bapiret2-type    = 'E'.
```

```
ls_bapiret2-id      = lc_mesg_cls.
```

```
ls_bapiret2-number  = '000'.
```

```
ls_bapiret2-message_v1 = TEXT-001.
```

\* Get the message text

```
MESSAGE ID lc_mesg_cls TYPE ls_bapiret2-type NUMBER ls_bapiret2-number
```

```
  WITH ls_bapiret2-message_v1 INTO ls_bapiret2-message.
```

```
APPEND ls_bapiret2 TO ct_return.
```

\* Raise the exception

```
RAISE abort.
```

ENDIF.

ENDMETHOD.